NPS72-86-002CR

NAVAL POSTGRADUATE SCHOOL
Monterey, California

CONTRACTOR REPORT

USING PHOENICS COMPUTER CODE FOR
TRANSIENT ONE-DIMENSIONAL METAL
COMBUSTION IN STEAM

Zeev Shavit

August 1986

Approved for public release; distribution unlimited

Prepared for:

Naval Surface Weapons Center/WOL
Code R10A
Silver Springs, MD 20910

NAVAL POSTGRADUATE SCHOOL
Monterey, California

RADM R. C. Austin
Superintendent

D. A. Schrady
Provost

The work reported herein was carried out for the Naval Postgraduate School by Mr. Zeev Shavit under Contract N62271-86-M-0217. The work presented in this report is in support of "Underwater Shaped Charges" sponsored by the Naval Surface Weapons Center. This work provides the implementation of the one dimensional transient model for metal combustion in steam in PHOENICS program. The model account for ignition and combustion stages. The project at the Naval Postgraduate School is under the cognizance of Distinguished Professor Allen E. Fuhs who is principal investigator.

Prepared by:

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS NONE
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution Unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NPS72-86-002CR	5 MONITORING ORGANIZATION REPORT NUMBER(S) NPS72-86-002CR
6a NAME OF PERFORMING ORGANIZATION Zeev Shavit	6b OFFICE SYMBOL (If applicable) 72
7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c. ADDRESS (City, State, and ZIP Code) Research Contractor Naval Postgraduate School Monterey, CA 93943-5100	7b. ADDRESS (City, State, and ZIP Code) Space Systems Academic Group Monterey, CA 93943-5100
8a NAME OF FUNDING/SPONSORING ORGANIZATION Naval Surface Weapons Center/WOL	8b OFFICE SYMBOL (If applicable) NSWC/WOL
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code) Naval Surface Weapons Center White Oak Laboratories Silver Springs, MD 20910	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

USING PHOENICS COMPUTER CODE FOR TRANSIENT ONE-DIMENSIONAL METAL COMBUSTION IN STEAM

12 PERSONAL AUTHOR(S)
ZEEV SHAVIT

13a. TYPE OF REPORT	13b TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Contractor Report	FROM Aug. 85 TO Aug. 86	AUGUST 1986	60

16. SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Metal Combustion, Combustion in Steam, Transient Droplet Combustion

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

In the previous report, presented in Reference 1, we considered the governing equations and physical parameters for metal droplet combustion in steam. A one dimensional (spherical) time-dependent model for burning process was proposed. In present report, we describe the implementation of that model in the general purpose computer code PHOENICX. This code can treat two phase transient flows in one or more space dimensions. Although we do not dwell on the specific of PHOENICS in this report, we describe in some detail the manner in which our model is incorporated into the code. The programming procedures are described via flow charts, list of variables and FORTRAN listing.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL ALLEN E. FUHS, Chairman	22b. TELEPHONE (Include Area Code) (408)646-2948 / 22c. OFFICE SYMBOL 72

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986—606-24

ABSTRACT

In the previous report, presented in Ref. 1, we considered the governing equations and physical parameters for metal droplet combustion in steam. A one dimensional (spherical) time-dependent model for burning process was proposed. In present report, we describe the implementation of that model in the general purpose computer code PHOENICS. This code can treat two phase transient flows in one or more space dimensions. Although we do not dwell on the specific of PHOENICS in this report, we describe in some detail the manner in which our model is incorporated into the code. The programming procedures are described via flow charts, list of variables and FORTRAN listing.

ACKNOWLEDGEMENT

The author wishes to express his gratitude to Professor Allen Fuhs for providing the opportunity to perform this work during my sabbatical, and for his support in participation as well as for the many valuable discussions.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

A	constant.
a	velocity of sound.
A_{ci}	internal cloud surface.
A_s	effective contact surface between two phases.
A_w	metal surface area.
B	constant in Eq. 6-34.
c_j	mass fraction concentration of species j.
c_{p_j}	species heat capacity.
c_{p_g}	gas heat capacity.
D_{jK}	binary diffusion coefficient.
D_{j^0}	mixture diffusion coefficient of species j.
E	activation energy.
f	mixture fraction.
F_p	friction force.
h	enthalpy.
H_{fu}	metal heat of combustion.
h_{sf}	solid to fluid latent heat.
h_{fg}	fluid to gas latent heat.
I	radiation intensity.
i	mass stochiometric constant.
k	heat conductivity.
k_{1j}	the k^{th} order coefficient in a polynomial to describe a temperature dependent property of the j^{th} species.
l	effective cloud particle thickness for radiation.
N	number of radiated particles per unit volume.
Nu	Nusselt number.
M	Mach number.
M_j	species molecular weight.
M_{12}	momentum transfer from phase 1 to phase 2.
p	pressure.
Q'''	heat source.
q''	heat flux.
R	specific gas constant.
r	radial distance.
$\mathcal{R}$	phase volume fraction

T	temperature.
s_i	ratio of internal cloud to metal surface area.
s_o	ratio of external cloud to metal surface area.
u	radial velocity.
w''_j	mass source of species j.
α	absorption coefficient.
β	flow parameter in stagnation region.
ε	emissivity.
$\in$	extinction coefficient.
λ	convection heat transfer coefficient.
Ω_{jk}	effective molecular diffusion cross section parameter.
μ	viscosity.
ρ	density.
γ	specific heat ratio.
σ	Boltzmann coefficient.
δ	covered oxide thickness layer.
χ	mole fraction.
Υ	stoichiometric mole coefficient.
Θ	nondimensional temperaure.

Subscripts

$_0$	initial.
$_1$	gas and metal phase.
$_2$	oxide phase.
$_c$	convection or cloud.
$_{fg}$	fluid to gas.
$_{fu}$	fuel.
$_g$	gas.
$_H$	hydrogen.
$_m$	melting.
$_M$	metal.
$_{ox}$	oxide.
$_p$	particle.
$_r$	wall radiation.
$_t$	total.
$_s$	solid.
$_{sf}$	solid to fluid.

st	stoichiometric.
wv	water vapor.

Superscript

^{0}	stagnation.

1. INTRODUCTION

Previous studies of fast metal combustion such as exploding wire or under water shaped charge have demonstrated that a theoretical description of the process calls for constructing a transient model rather than existing quasi-steady models. In some particular cases of volatile metal combustion of large droplets (larger than 80 micron), radiation heat exchange between the droplet surface and the surrounding oxide cloud cannot be ignored. Special care was taken in Ref. 1 to properly account for the oxide cloud and the metal surface radiation. Oxide cloud formation, convection and deposition can be described by the transport equation of the second phase. The metal is treated as a single phase whether it is liquid or vapor. This is accomplished through a judiciuos choice of enthalpy which accounts for vapor quality and latent heat of vaporization. Metal-vapor interface location is found indirectly by searching for a point where the temperature is equal to the saturated vapor temperature corresponding to the local metal partial pressure, and metal vapor quality is zero.

In Ref. 1, we described the equations for spherical, one dimensional, transient combustion an including oxide as a second phase. In order to imbed these equations in a computer program, we will use the general purpose code PHOENICS , Ref. 2. Since this code is not restricted to one spatial dimension, once the feasibility of the 1-D model in this phase of the work has been demonstrated, it may be extended to 2-D (axisymmetric using r and θ as spatial variables) in a straightforward manner.

2. THE GENERAL TRANSPORT EQUATION IN PHOENICS

The underlying idea of PHOENICS difference scheme is that all transport equations are re-cast in a uniform format- the "general transport equation". The formating includes the energy conservation equation. The governing equations for our model are conventionally written as written as follows:

Energy equation phase 1:

$$\rho_1 Dh_1/Dt = \nabla(K_1 \nabla T_1) + Q''_1 \tag{1}$$

Energy equation phase 2:

$$\rho_2 Dh_2/Dt = \nabla(K_2 \nabla T_2) + Q''_2 \tag{2}$$

Species mass fraction:

$$\rho_j Dc_j/Dt = \nabla(\rho_1 D_{j0} \nabla c_j) + w''_j \tag{3}$$

Momentum phase 1 :

$$\rho_1 DV_1/Dt = \nabla(\mu_1 \nabla V_1) + \mathbf{M}_{21} \tag{4}$$

Momentum phase 2 :

$$\rho_2 DV_2/Dt = \nabla(\mu_2 \nabla V_2) + \mathbf{M}_{12} \tag{5}$$

The continuity equation for each phase:

$$\partial(\rho_1 \mathscr{R}_1)/\partial t + (\rho_1 V_1 \mathscr{R}_1) = m_{21} \tag{6}$$

$$\partial(\rho_2 \mathscr{R}_2)/\partial t + (\rho_2 V_2 \mathscr{R}_2) = -m_{21} \tag{7}$$

As shown by Spalding (Ref. 2), these equations can be presented in th following generalized form, utilizing in addition to the property Φ, source terms S''_Φ and exchange coefficients Γ_Φ

$$\rho_j D\Phi_j/Dt = \nabla(\Gamma_\Phi \nabla \Phi_j) + S''_{\Phi_j} \tag{8}$$

Where:

The variable $\Phi_j = h_1, h_2, c_j, V_1, V_2$ (8a)

The source $S'' = Q''_1, Q''_2, w''_j, \mathbf{M}_{12}, \mathbf{M}_{21}$ (8b)

The exchange coefficient:

$$\Gamma_\Phi = K/c_p, \rho_1 D_{j0}, \mu_1, \mu_2 \tag{8c}$$

This generalization enables the use of a single numerical scheme for all transport equations. PHOENICS is built in three main parts described in Fig. 1.

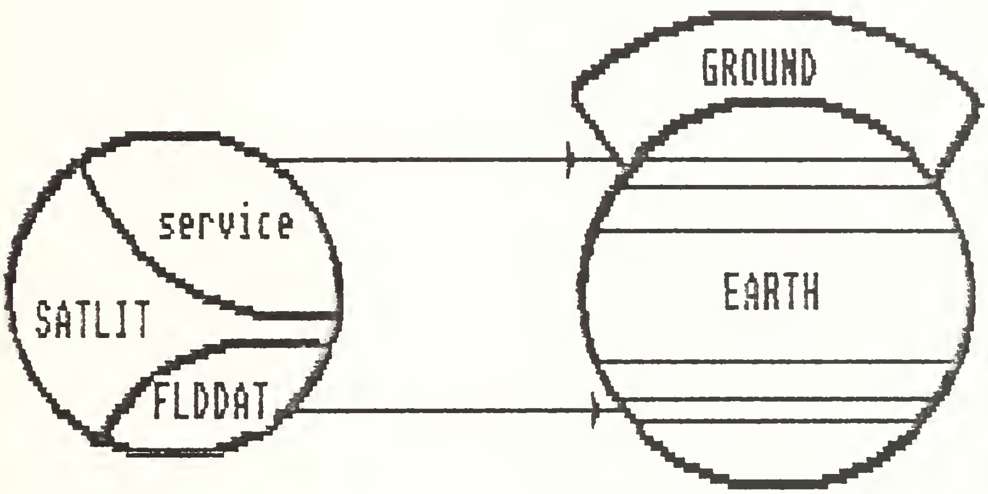


FIg. 1 The various subprograms in PHOENICS.

EARTH	Include the numeric finite differences scheme and written as binary code.
SATLIT	The management program, where the problem is declared: steady/unsteady, parabolic or eliptic , boundary conditions, relaxation factors etc.
FLDDAT	Initial values definition for each variable Φ_j being used.
GROUND	For variables updating and sources definition.

3. CORRECTION CONDUCTION TERM IN ENERGY EQUATION

Special care should be excercised in recasting the energy equation into the general format. The problem arises from the heat conduction term which should be a divergence of K T. Since the generalizated energy variable is enthalpy rather than temperature, the heat flux is related indirectly to the enthalpy. In PHOENICS this difficulty is resolved by assuming a constant (temperature and composition independent) value of c_p.

This approximation is inacceptable for our model. In the sequel, we dwell on the reasons for that and on the proposed remedy.

There are three major factors that prohibit the assumption of constant c_p. These are:

(a) Large temperature gradients.

(b) Large composition gradients (especially in the combustion region and around the liquid metal/vapor interface).

(c) The conductive heat flux constitutes a non-negligible fraction of the total heat flux (conduction + convection + radiation).

The remedy for this problem is to introduce a suitable source term into the PHOENICS-generalized energy equation. This source term is just the difference between the correction heat flux divergence and the enthalpy flux divergence. For our model, the appropriate energy source correction term is:

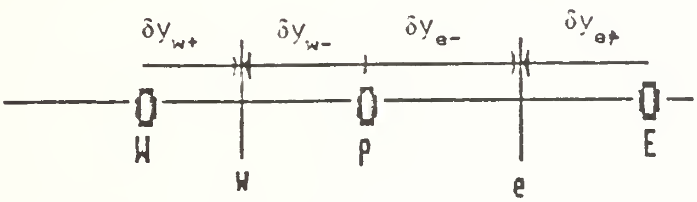
$$
\begin{aligned}
S''_{corr} = {} & \partial[(K/c_{p_0})[c_{p_M}(\partial c_M/\partial y) + c_{p_H}(\partial c_H/\partial y) + c_{p_{wv}}(\partial c_{wv}/\partial y)]\,(T-298)]/\partial y \\
& - \partial[(K/c_p)h^{\circ}_{wv}(\partial c_{wv}/\partial y)]/\partial y \\
& - \partial[(K/c_p)[(1-x)h_{Mfg}(\partial c_M/\partial y) - h_{Mfg}c_M(\partial x/\partial y)]]/\partial y
\end{aligned} \qquad (9)
$$

Note that last term in Eq. 9 vanishes except in the transition region between metal liquid and vapor, where x is a fraction number.

In the case of a 1-D configuration, finite- difference gradient approximation is available in PHOENICS and described in Ref. 3.

$$\partial[(\Gamma)\partial h/\partial y] \equiv (h_E-h_P) / [\ \delta y_{e-}/\Gamma_P + \delta y_{e+}/\Gamma_E]$$
$$- (h_P-h_W) / [\ \delta y_{w-}/\Gamma_P + \Delta y_{w+}/\Gamma_W] \qquad (10)$$

Where p is the center node and E and W are its sides neighbors, as described below:



We used this difference scheme in computing the corrective energy terms.

4. VARIABLE NAMES AND FLOWCHART

In this chapter we wil present the metal combustion variables used in PHOENICS and the implementation of the model equations as they appear in chapter 6 of Ref. 1.. The definition of the problem and its boundary condition was declared in SATLIT. The initial conditions were set in FLDDAT. The various sources, the iterations procedures of surface and oxide cloud location together with the other combustion model were inserted in GROUND. The reader can follow the flow chart together with using the FORTRAN list in the appendix.

NAME IN THE PROGRAM	NAME OF VARIABLE IN EQUATION	APPEAR IN EQUATION NO.	*
GAI	s_i	6-27	C
GA0	s_o	6-28	C
GAP	α_p	6-26,30	
GAQ	E/R	6-34	
GCDT	w''_m	6-34	C
GCP	c_{p_j}	6-51c	C
GCPI	c_{p_g}	6-26C	
GCPOX	$c_{p_{ox}}$		
GCP0	c_p0	6-51c	
GCP1	c_p1	6-51c	
GCP2	c_p2	6-51c	
GCP3	c_p3	6-51c	
GCP4	c_p4	6-51c	
GC1	c_M	6-39	C
GC2	c_{wv}	6-39	C
GC3	c_H	6-39	C

GC1MIN	N_{min}	parameter for radiation beginning	
GD	D_{j^0}	6-49	C
GDD	$D_{pm_{jk}}$	6-48	C
GD1	ρ_1	6-55	C
GD2	ρ_2		
GEG	ε_g	6-26,30	
GEP	ε_p	6-25	C
GEW	ε_w	6-26,30	
GF	c_M/c_{wv}	6-35a,b	
GFCW	F_{c-w}	6-29	
GFG	i_H	6-37a	
GFOX	i_{ox}	6-37b	
GGAMH1	k/c_{p_g}		
GHFGM	h_{Mfg}	6-4	
GHFOX	h_{oxfg}	6-32	
GHFU	H_{fu}	6-5	
GHOXS	Q''_{12}	6-6a	C
GHOXV	Q''_{13}	6-6b	C
GHOX0	h_{ox}^0	6-33a	
GH1M	h_{M1}	6-9	C
GH3M	h_{M3}	6-10	C
GK	k_j	6-51a	C
GKA	B	6-34	
GKMF	K_{Mf}		

GK0	k_{j0}	6-51a	
GK1	k_{j1}	6-51a	
GK2	k_{j2}	6-51a	
GK3	k_{j3}	6-51a	
GK4	k_{j4}	6-51a	
GL	l	6-25	
GLSOX	h_{oxsf}	6-6b	
GM	μ_j	6-51a	C
GMDT	w_{ox}''	6-37a,b	C
GMM	M_j/M_k	6-52	
GM0	μ_{j0}	6-51,b	
GM1	μ_{j1}	"	
GM2	μ_{j2}	"	
GM3	μ_{j3}	"	
GM4	μ_{j4}	"	
GN	N_i	6-23	C
GNMI	$N_{i_{min}}$		
GNN	N	6-24	C
GPAL	p_M	6-18	
GPT	p		
GRC	r_c	6-29	
GRCI	r_{ci}	6-27,28	
GRCo	r_{co}	"	
GRFC	$q_{ox_{net}}$	6-26	C
GROF	ρ_{Mf}	6-21	

GROG	ρ_{Mg}	6-21	
GRO0	ρ_{ox}	6-23	
GROU	$\rho_1 u_1$	6-1,39,45	
GROUCM	$\rho_1 u_1 \partial c_M/\partial r$	6-39	
GROUOX	$\rho_1 u_1 \partial c_{wv}/\partial r$	6-39	
GRW	q''_{wnet}	6-30	C
GR2	$\mathscr{R}_2$	6-42,46	
GSIG	σ	6-26,30	
GSUMD	$\Sigma\chi_k/D_{kj}$	6-49	
GSUMF	$\Sigma\chi_k\Omega_{jk}$	6-53,54	
GSUMK	K	6-54	C
GSUMN	$\Sigma N_i \Delta V_i$	6-24	
GSUMU	μ	6-53	C
GSUMV	$\Sigma\Delta V_i$	6-24	
GSXM2		term in Eq. 9	
GSXM1		"	
GTEMP	T_1	6-17	C
GTEMPO	T_2	6-33a	
GTG	T_g	at steam	
GTOX	T_{ox}	at cloud	
GTW	T_w	at x=0	
GVISC	μ		
GXMET	x	6-19	C
GXMETA	x'	assumed for iteration	

* Calculated in that equation.

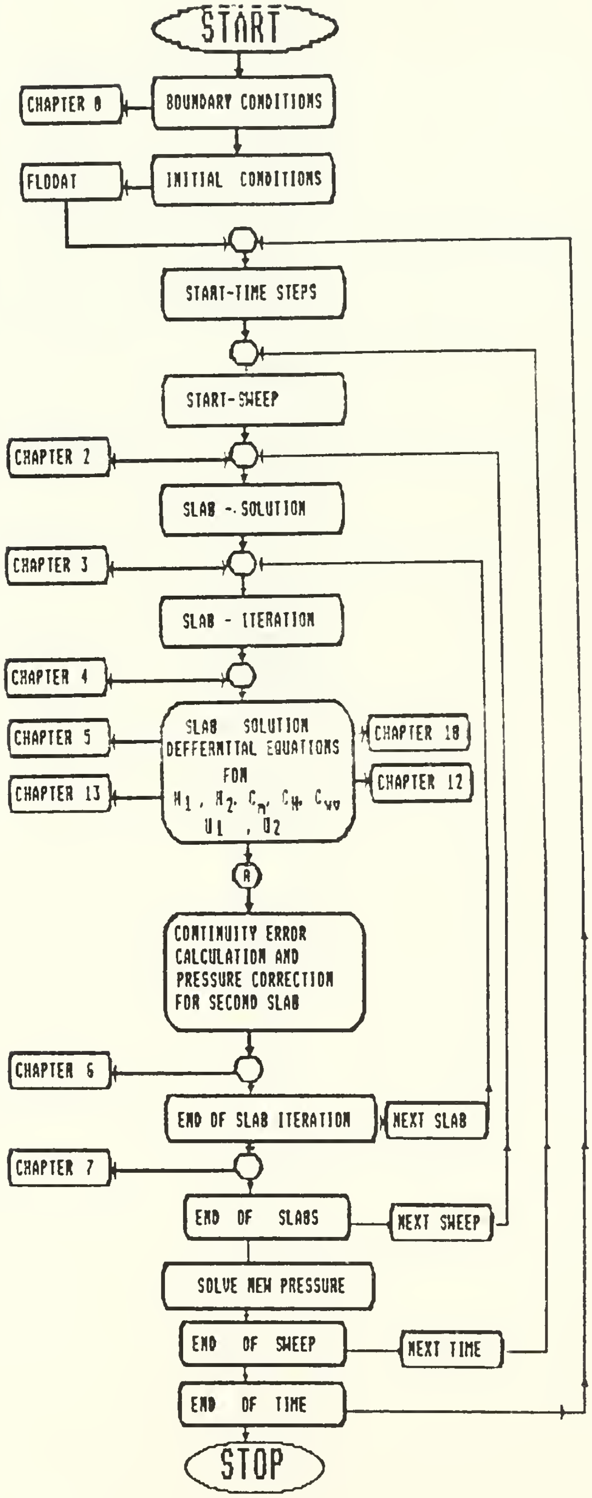


Fig. 2 General flow-chart of the model in PHOENICS

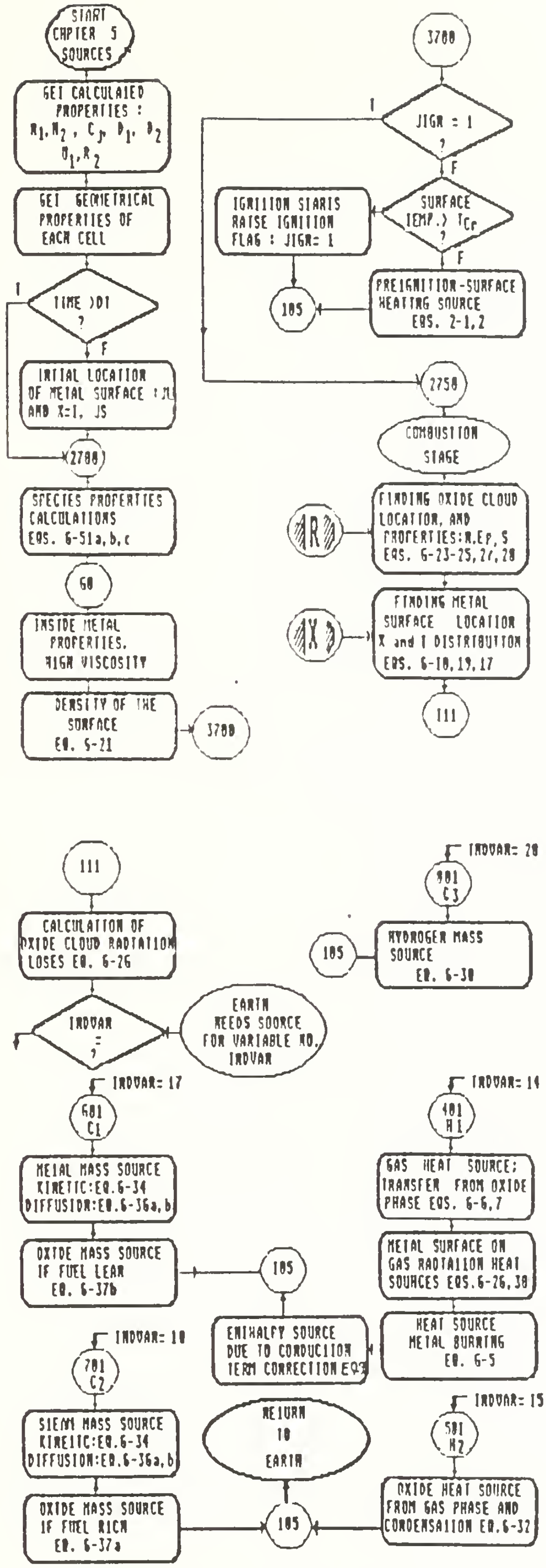


Fig. 3 Chapter 5 in GROUND for the various sources in the model

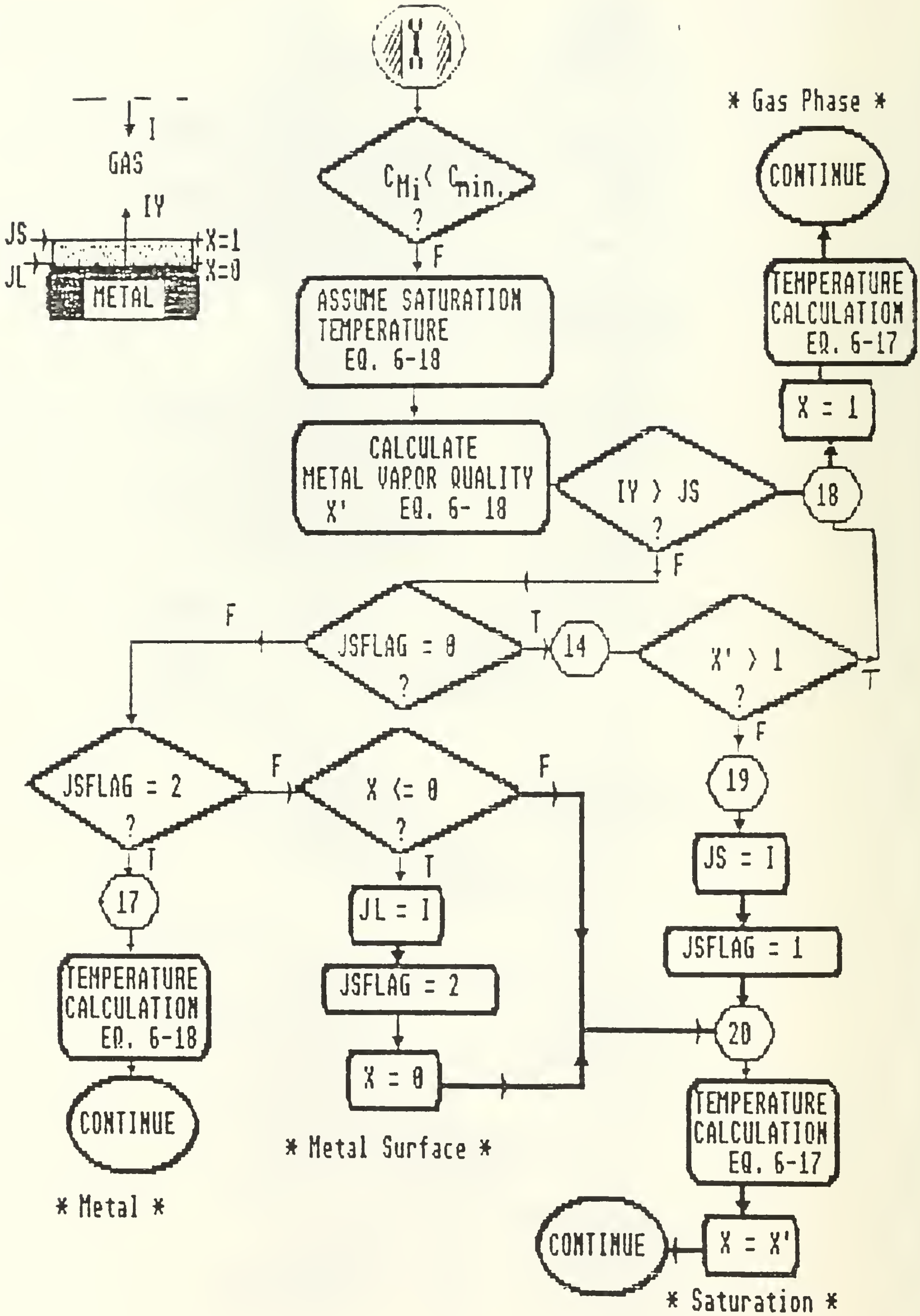


Fig. 4 Metal vapor distribution and surface location

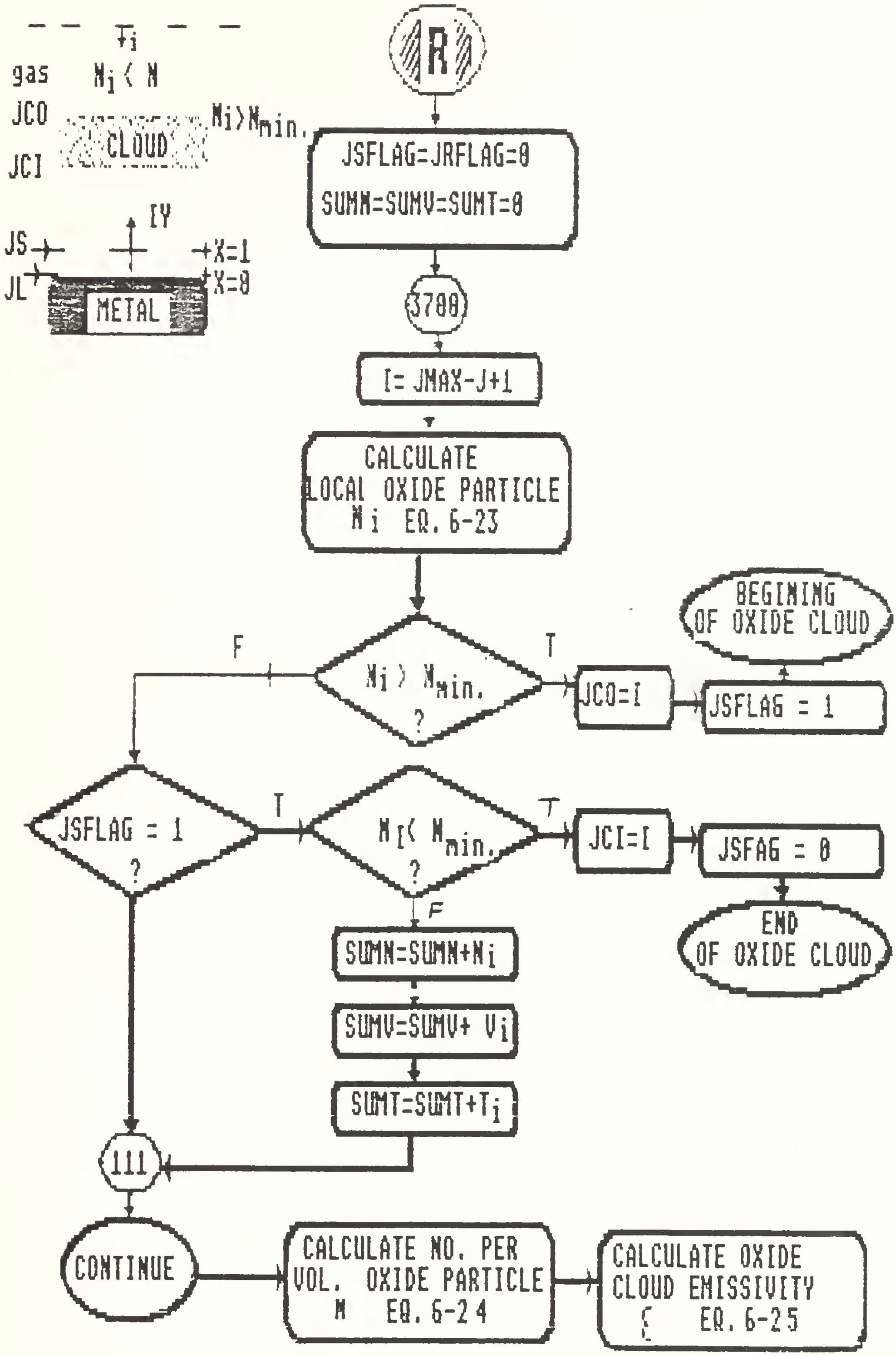


Fig. 5 Oxide cloud location and cloud emissivity

5. SUMMARY

We presented a prescription for embedding the 1-D droplet combustion model that had been proposed in a Ref. 1 into the PHOENICS code. Extensive programming modifications are required, primarily due to the introduction of special constructed (finite- difference) source terms. These terms are of two distinct types. The corrective type was inserted into the energy equation in order to offset errors introduced by the general equation format of PHOENICS, which is based on an assumption of constant c_p, that does not hold in the present model.

The second type is true mass and energy source terms. The mass source is due to the progress of chemical reactions. The energy source is due to radiative heat transfer.

The debugging phase has just about been completed at the time this report was written. We regard the 1-D model as a test case for droplet combustion in steam. Most- but not all - the physical factors are represented in the model. Specially, oxide deposition on a moving droplet can be modeled in a 2-D (axisymmetric) computation.

Once the 1-D test computations prove statisfactory, the road is clear to proceed with the 2-D model.

In this appendix we will pesent the FORTRAN listing of PHOENICS which includes our model for metal combustion in steam.

```
C$DIRECTIVE**MAIN
C     *FILE NAME: MODGRD.FTN
C     *INCLUDED SUBROUTINES: THE MODELS OF MAIN, GROUND & STRIDE.
C     *DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983).
C     *SATELLITE FILE NAME: MODSTL.FTN
      COMMON/ISHIFT/III(57),NFMAX
C SET F-ARRAY DIMENSION AS NEEDED, & SET NFMAX ACCORDINGLY.
      COMMON F(25000)
      NFMAX=25000
      CALL MAIN1
      STOP
      END
C$DIRECTIVE**GROUND
      SUBROUTINE GROUND(IRN,ICHAP,ISTP,ISWP,IZED,INDVAR)
C$INCLUDE 9,CMNGUSSI.FTN/G
      LOGICAL LOGIC1,LOGIC
      DIMENSION LOGIC(100)
      COMMON/LDATA/LOGIC1(309)
      EQUIVALENCE (LOGIC(1),LOGIC1(210))
      DIMENSION INTGR(100)
      COMMON/IDATA/INTGR1(194)
      EQUIVALENCE (INTGR(1),INTGR1(95))
      DIMENSION RE(100)
      COMMON/RDATA/RE1(421)
      EQUIVALENCE (RE(1),RE1(322))
      COMMON/BOUND/LOCREG(60),
     &TR1,CP1R1(7),VP1R1(7),CP2R1(5),VP2R1(5),CPNR1(5),VPNR1(5),
     &TR2,CP1R2(7),VP1R2(7),CP2R2(5),VP2R2(5),CPNR2(5),VPNR2(5),
     &TR3,CP1R3(7),VP1R3(7),CP2R3(5),VP2R3(5),CPNR3(5),VPNR3(5),
     &TR4,CP1R4(7),VP1R4(7),CP2R4(5),VP2R4(5),CPNR4(5),VPNR4(5),
     &TR5,CP1R5(7),VP1R5(7),CP2R5(5),VP2R5(5),CPNR5(5),VPNR5(5),
     &TR6,CP1R6(7),VP1R6(7),CP2R6(5),VP2R6(5),CPNR6(5),VPNR6(5),
     &TR7,CP1R7(7),VP1R7(7),CP2R7(5),VP2R7(5),CPNR7(5),VPNR7(5),
     &TR8,CP1R8(7),VP1R8(7),CP2R8(5),VP2R8(5),CPNR8(5),VPNR8(5),
     &TR9,CP1R9(7),VP1R9(7),CP2R9(5),VP2R9(5),CPNR9(5),VPNR9(5),
     &TR10,CP1R10(7),VP1R10(7),CP2R10(5),VP2R10(5),CPNR10(5),VPNR10(5)
      DIMENSION TCVREG(350)
      EQUIVALENCE (TCVREG(1),TR1)
C$INCLUDE 9,GUSSEQUI.FTN/G
      LOGICAL CARTES,POLAR,SPDATA,SKEW,TWODYZ,ONEDZ,STOVAR(25),
     &   SOLVAR(25),PRINT(25),RESID(25),CMPRSS,CONEMU,LSP1,
     &   CONRHO,EMDOT,ONEPHS,INCORE(10),ISAVED,SAVEI,SAVEM,
     &   RESTRT,XCYCLE,MONITR,REGION(10),STEADY,WHOLEP,SLABPP,
     &   RAIN,BLOCKZ,PWSTAG,RUN(30),PLOT,RESMAP,FLAG,BLOCK,
     &   TEST,CONC1(4),DISTIL
      LOGICAL SPRESS,PARAB,DONACC,OVERLY,SACC,
     &   GUSSIE,CATHY,CONNIE,CORA,ESTER,FLASH,FLORA,
     &   FOCS,GENMIX,HESTER,PICALO,PLANT,SPLASH,HELGA,
     &   TACT,TIBALT,TOPSI,PAMELA,TABLES,WSTAG,
     &   CONMOD,GROSTA,SUBPST,SUBWGR
      EQUIVALENCE (CARTES,LOGIC1(1)),(POLAR,LOGIC1(2)),
     &   (SPDATA,LOGIC1(3)),(SKEW,LOGIC1(4)),(TWODYZ,LOGIC1(5)),
     &   (ONEDZ,LOGIC1(6)),(STOVAR(1),LOGIC1(7)),
     &   (SOLVAR(1),LOGIC1(32)),(PRINT(1),LOGIC1(57)),
     &   (RESID(1),LOGIC1(82)),(CMPRSS,LOGIC1(107)),
     &   (CONEMU,LOGIC1(108)),(LSP1,LOGIC1(109)),
     &   (CONRHO,LOGIC1(110)),(EMDOT,LOGIC1(111)),
     &   (ONEPHS,LOGIC1(112)),(INCORE(1),LOGIC1(113))
      EQUIVALENCE(ISAVED,LOGIC1(123)),(SAVEI,LOGIC1(124)),
     &   (SAVEM,LOGIC1(125)),(RESTRT,LOGIC1(126)),
     &   (XCYCLE,LOGIC1(127)),(MONITR,LOGIC1(128)),
     &   (REGION(1),LOGIC1(129)),(STEADY,LOGIC1(139)),
     &   (WHOLEP,LOGIC1(140)),(SLABPP,LOGIC1(141)),
     &   (RAIN,LOGIC1(142)),(BLOCKZ,LOGIC1(143)),
     &   (PWSTAG,LOGIC1(144)),(RUN(1),LOGIC1(145)),
     &   (PLOT,LOGIC1(175)),(RESMAP,LOGIC1(176)),
     &   (FLAG,LOGIC1(177)),(BLOCK,LOGIC1(178)),(TEST,LOGIC1(179))
      EQUIVALENCE(CONC1(1),LOGIC1(206))
      EQUIVALENCE (SPRESS,LOGIC(100)),(PARAB,LOGIC(99)),
     &   (DONACC,LOGIC(98))
      EQUIVALENCE (OVERLY,LOGIC(96)),(SACC,LOGIC(95)),
     &   (WSTAG,LOGIC(94))
```

```
      EQUIVALENCE (DISTIL,LOGIC(90))
      EQUIVALENCE (PAMELA,LOGIC(67)),(TOPSI,LOGIC(66)),
     &   (TIBALT,LOGIC(65)),(TACT,LOGIC(64)),(HELGA,LOGIC(63)),
     &   (SPLASH,LOGIC(62)),(PLANT,LOGIC(61)),(PICALO,LOGIC(60)),
     &   (HESTER,LOGIC(59)),(GENMIX,LOGIC(58)),(FOCS,LOGIC(57)),
     &   (FLORA,LOGIC(56)),(FLASH,LOGIC(55)),(ESTER,LOGIC(54)),
     &   (CORA,LOGIC(53)),(CONNIE,LOGIC(52)),(CATHY,LOGIC(51)),
     &   (GUSSIE,LOGIC(50))
      EQUIVALENCE (TABLES,LOGIC(48)),(CONMOD,LOGIC(47)),
     &   (GROSTA,LOGIC(46)),(SUBPST,LOGIC(45))
      EQUIVALENCE (SUBWGR,LOGIC(43))
      INTEGER FSTEP,FSWEEP,TSTSWP,ITAB(8),MTABVR(8)
      DIMENSION ISPCSO(25),LITER(25)
      EQUIVALENCE (NX,INTGR1(1)),(NY,INTGR1(2)),
     &   (NZ,INTGR1(3)),(ISPCSO(1),INTGR1(4)),
     &   (NREGN,INTGR1(29)),(NPHI,INTGR1(30)),
     &   (LITKE,INTGR1(31)),(LITHYD,INTGR1(32)),
     &   (LITH,INTGR1(33)),(LITCNC,INTGR1(34)),
     &   (LITSLB,INTGR1(35)),(NRUN,INTGR1(36)),
     &   (LITER(1),INTGR1(37)),(FSTEP,INTGR1(62)),
     &   (FSWEEP,INTGR1(63)),(LSTEP,INTGR1(64))
      EQUIVALENCE (LSWEEP,INTGR1(65)),(NPRINT,INTGR1(66)),
     &   (IERRP,INTGR1(67)),(IMAXP,INTGR1(68)),
     &   (IEMU1,INTGR1(69)),(IXMON,INTGR1(70)),
     &   (IYMON,INTGR1(71)),(IZMON,INTGR1(72)),
     &   (KEMU,INTGR1(73)),(KMAIN,INTGR1(74)),
     &   (KINDEX,INTGR1(75)),(KGEOM,INTGR1(76)),
     &   (KINPUT,INTGR1(77)),(KSODAT,INTGR1(78))
      EQUIVALENCE(KFLAG,INTGR1(79)),(KCOMPF,INTGR1(80)),
     &   (KSORCE,INTGR1(81)),(KLES1D,INTGR1(82)),
     &   (KLES2D,INTGR1(83)),(KLES3D,INTGR1(84)),
     &   (KCOMPP,INTGR1(85)),(KADJST,INTGR1(86)),
     &   (KFLUX,INTGR1(87)),(KSHIFT,INTGR1(88)),
     &   (KOUTPT,INTGR1(89)),(KDIF,INTGR1(90)),
     &   (KCOMPU,INTGR1(91)),(KCOMPV,INTGR1(92)),
     &   (KCOMPW,INTGR1(93)),(KCOMPR,INTGR1(94))
      EQUIVALENCE (KELIN,INTGR(100)),(MEANDF,INTGR(99)),
     &   (NUMCLS,INTGR(98)),
     &   (IRHO1,INTGR(97)),(IRHO2,INTGR(96)),
     &   (IZW1,INTGR(95)),(IZW2,INTGR(94)),
     &   (MGRID,INTGR(93)),(KWALL,INTGR(92)),
     &   (IZPR1,INTGR(91)),(IZPR2,INTGR(90)),
     &   (ISTPR1,INTGR(89)),(ISTPR2,INTGR(88)),
     &   (NTPRIN,INTGR(87))
      EQUIVALENCE (IMDOT,INTGR(86)),(IHSAT,INTGR(85)),
     &   (ICFIP,INTGR(84)),(NTABLE,INTGR(83)),(NTABVR,INTGR(82)),
     &   (LINTAB,INTGR(81)),(NPRTAB,INTGR(80)),(NMON,INTGR(79)),
     &   (ITAB(1),INTGR(71)),(MTABVR(1),INTGR(63)),
     &   (NZPRIN,INTGR(62)),
     &   (NXPRIN,INTGR(61)),(NYPRIN,INTGR(60)),
     &   (LDISTL,INTGR(59)),(TSTSWP,INTGR(58)),
     &   (KDBEXP,INTGR(57)),(KDBRHO,INTGR(56)),
     &   (KDBMDT,INTGR(55))
      EQUIVALENCE (ILOOP1,INTGR(53)),(ILOOPN,INTGR(52)),
     &   (IPBP,INTGR(51)),(IZPRF,INTGR(50)),
     &   (IZPRL,INTGR(49)),(ISTPRF,INTGR(48)),
     &   (ISTPRL,INTGR(47)),(KDBGEN,INTGR(46)),
     &   (IVELF,INTGR(45)),(IVELL,INTGR(44)),
     &   (IKEF,INTGR(43)),(IKEL,INTGR(42)),
     &   (IENTF,INTGR(41)),(IENTL,INTGR(40)),
     &   (ICNCF,INTGR(39)),(ICNCL,INTGR(38)),
     &   (NVEL,INTGR(37)),(NKE,INTGR(36)),
     &   (NENT,INTGR(35)),(NCNC,INTGR(34)),
     &   (NZSTP,INTGR(33)),(NPRMNT,INTGR(32))
      DIMENSION SPARE1(20),XUDIST(30),YVDIST(30),ZWDIST(50),
     &   SIGMA(25),CRIT(25),DTFALS(25),RESREF(25),
     &   TITLE(25),FIINIT(25),TFRAC(30)
      EQUIVALENCE (SPARE1(1),RE1(1)),
     &   (TFRAC(1),RE1(21)),(XULAST,RE1(51)),
     &   (YVLAST,RE1(52)),(ZWLAST,RE1(53)),
     &   (XUDIST(1),RE1(54)),(YVDIST(1),RE1(84)),
     &   (ZWDIST(1),RE1(114)),(SIGMA(1),RE1(164)),
```

```
 &    (CRIT(1),RE1(189)),(DTFALS(1),RE1(214)),
 &    (RESREF(1),RE1(239)),(EMU1,RE1(264)),
 &    (RHO1,RE1(265)),(RHO2,RE1(266)),
 &    (TLAST,RE1(267)),(CFIPS,RE1(268))
  EQUIVALENCE (AMDOT,RE1(269)),(FIINIT(1),RE1(270)),
 &    (RELAXP,RE1(295)),(TITLE(1),RE1(296)),
 &    (DT,RE1(321)),
 &    (RINNER,ZWDIST(50)),(SNALFA,ZWDIST(49)),
 &    (PBAR,ZWDIST(48))
  EQUIVALENCE (SLOEMU,RE(100)),(SLORHO,RE(99)),(RLXRHO,RE(98)),
 &    (RHOMAX,RE(97)),(RHOMIN,RE(96)),
 &    (EMUMAX,RE(95)),(EMUMIN,RE(94)),
 &    (TKEMAX,RE(93)),(TKEMIN,RE(92)),
 &    (RLXPZ,RE(91)),
 &    (ABUOY,RE(90)),(HREF,RE(89)),
 &    (AGRAVX,RE(88)),(AGRAVY,RE(37)),
 &    (AGRAVZ,RE(86)),(ATIME,RE(85))
  EQUIVALENCE (RLXPXY,RE(84)),
 &    (EPSMAX,RE(83)),(EPSMIN,RE(82)),
 &    (ARHO1,RE(81)),(BRHO1,RE(80)),(CRHO1,RE(79))
  EQUIVALENCE (AZW2,RE(73)),(BZW2,RE(74)),
 &    (PINT,RE(75)),(CZW2,RE(76)),
 &    (DZ,RE(78)),(ZWL,RE(77)),(ZW2M1T,RE(72)),
 &    (VELMIN,RE(70)),(VELMAX,RE(69)),
 &    (FALCOM,RE(68)),(CMDOT,RE(67)),
 &    (RLXMDT,RE(66)),(AMDTMX,RE(65)),(AMDTMN,RE(64)),
 &    (RADMAX,RE(63)),(RADMIN,RE(62)),
 &    (PRESS,RE(61)),(CP1,RE(60)),
 &    (CP2,RE(59)),(CP3,RE(58)),(FSTOIC,RE(57)),
 &    (ARRCON,RE(56)),(PREEXP,RE(55)),(CA1I,RE(54))
  EQUIVALENCE (CA2I,RE(53)),(CA13,RE(52)),(CA23,RE(51)),
 &    (H1SAT,RE(50)),(H2SAT,RE(49)),
 &    (EMULAM,RE(48))
  EQUIVALENCE (TWPRCN,RE(42)),(PSATEX,RE(41))
  EQUIVALENCE(ELEXP,RE(39)),(EWALL,RE(38)),
 &    (SIGE,RE(37)),(SIGK,RE(36)),(TAUDK,RE(35))
  DIMENSION LOC1(6),LOC2(6),LOC3(6),LOC4(6),LOC5(6),
 &    LOC6(6),LOC7(6),LOC8(6),LOC9(6),LOC10(6)
  EQUIVALENCE (LOCREG(1),LOC1(1)),(LOCREG(7),LOC2(1))
  EQUIVALENCE (LOCREG(13),LOC3(1)),(LOCREG(19),LOC4(1))
  EQUIVALENCE (LOCREG(25),LOC5(1)),(LOCREG(31),LOC6(1))
  EQUIVALENCE (LOCREG(37),LOC7(1)),(LOCREG(43),LOC8(1))
  EQUIVALENCE (LOCREG(49),LOC9(1)),(LOCREG(55),LOC10(1))
  DIMENSION XUFRAC(30),YVFRAC(30),ZWFRAC(30),ENDIT(25)
  DIMENSION XFRAC(30),YFRAC(30),ZFRAC(30)
  EQUIVALENCE(XUDIST(1),XUFRAC(1),XFRAC(1)),
 &          (YVDIST(1),YVFRAC(1),YFRAC(1)),
 &           (ZWDIST(1),ZWFRAC(1),ZFRAC(1)),(LITCNC,LITC),
 &           (ENDIT(1),CRIT(1)),(RLXP,RELAXP)
  EQUIVALENCE (NPRMON,LITER(1))
CLUDE 9,NMLIST.FTN/G
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 STARTS:
----------------------------------------------------------------
++MEANING OF SUBROUTINE ARGUMENTS:
  IRN=RUN NUMBER; ICHAP=CHAPTER CALLED; ISTP=TIME STEP;
  ISWP=SOLUTION SWEEP; IZED=Z-SLAB; INDVAR: SEE CHAPTERS BELOW.
++USER-INTRODUCED VARIABLES & ARRAYS:
  TO AVOID CONFLICT WITH VARIABLE NAMES USED IN COMMON, ALL
  VARIABLES INTRODUCED BY THE USER SHOULD HAVE NAMES STARTING
  WITH 'G' IF REAL, 'J' IF INTEGER, AND 'G' OR 'J' IF LOGICAL.
  THUS GDZ(IZ) MIGHT BE A Z-INTERVAL ARRAY;
  GW1(IY,IX) A 2-D ARRAY FOR AXIAL VELOCITY; ETC.
  USER-GENERATED SUBROUTINES SHOULD BE NAMED CORRESPONDINGLY,EG
  SUBROUTINE GVISC(GTEMP,GCNC,GVSC), FOR COMPUTING VISCOSITY
  FROM CONCENTRATION & TEMPERATURE.
++GROUND-TO-EARTH CONNECTING SUBROUTINES:
 *USE GET(NAME,GARRAY,NY,NX) TO PUT VALUES OF VARIABLE NAMED
  'NAME' INTO ARRAY 'GARRAY' DIMENSIONED GARRAY(NY,NX).
 *USE SET(NAME,IXF,IXL,IYF,IYL,GARRAY,NY,NX) TO SET VARIABLE
  'NAME' TO GARRAY(IY,IX) OVER THE REGION: IXF-IXL & IYF-IYL.
 *USE PRNSLB(NAME) TO PRINT VARIABLE 'NAME' OVER X-Y PLANE.
 *USE ADD(NAME,IXF,IXL,IYF,IYL,TYPE,CM,VM,CVAR,VVAR,NY,NX)
```

```
C        TO ADD SOURCE TO VARIABLE NAMED 'NAME' (SEE CHAPTER 5).
C      *USE READIZ(IZED) IN CHAPTERS 1, 2, 8, & 9 TO ACCESS P1,...DM
C        & VOL,...AHDZ. (SEE FOOTNOTE TO LEGALITY TABLE)
C      *USE GET1D(NAME,GARRAY,NDIM) TO PUT VARIABLE NAMED 'NAME' IN
C        ONE-D ARRAY 'GARRAY' DIMENSIONED NDIM, THUS:
C        CALL GET1D(NAME,GNX,NX) FOR XG,...DXG & DIMENSION GNX(NX);
C        CALL GET1D(NAME,GNY,NY) FOR YG,...RV & DIMENSION GNY(NY);
C        CALL GET1D(NAME,GNZ,NZ) FOR ZG,...WGRID & DIMENSION GNZ(NZ).
C+++++LEGALITY TABLE FOR USE OF EARTH-CONNECTING SUBROUTINES:
C        ENTRIES IN TABLE GIVE CHAPTERS IN WHICH SUBROUTINES CAN BE
C        USED FOR VARIABLES IN LEFT-HAND COLUMN. (SUBROUTINE
C        STRIDE IS REGARDED AS BEING IN CHAPTER 3)
C        -------------------------------------------------------------
C        : VARIABLE:: GET &    :   SET  : ADD     : READIZ : GET1D :
C        :          ::  PRNSLB :        :         :        :       :
C        -------------------------------------------------------------
C        :P1 - RZ   ::    ALL   :  6 & 7 :   5     : 1,2,8,9:  NONE :
C        :P10 - RZH::3-7, 10-16:     3   :  NONE   :  NONE  :  NONE :
C        :VOL -AHDZ::    ALL   :     3   :  NONE   : 1,2,8,9:  NONE :
C        :D1DP      ::    NONE  :    10   :  NONE   :  NONE  :  NONE :
C        :D2DP      ::    NONE  :    11   :  NONE   :  NONE  :  NONE :
C        :MU1,MU1H :: 5,13-16   :    12   :  NONE   :  NONE  :  NONE :
C        :EXCO(L,H)::    NONE  :    13   :  NONE   :  NONE  :  NONE :
C        :CFP       ::     5    :    14   :  NONE   :  NONE  :  NONE :
C        :MDT       ::     5    :    15   :  NONE   :  NONE  :  NONE :
C        :HST1,HST2::  5 & 15   :    16   :  NONE   :  NONE  :  NONE :
C        :XG -WGRID::    NONE  :  NONE   :  NONE   :  NONE  :  ALL  :
C        -------------------------------------------------------------
C        NOTES ON ABOVE TABLE:
C      *IN CHAPTERS 1, 2, 8, & 9 VARIABLES P1...DM & GEOMETRY
C        VOL...AHDZ CAN BE ACCESSED BUT ONLY IN CONJUNCTION WITH
C        USE OF READIZ, THUS:
C        DO 1 IZED=1,NZ
C        CALL READIZ(IZED)
C     1 CALL GET(... AS REQUIRED..)
C      *GEOMETRY ACCESSED BY READIZ IS THAT AT INITIAL TIME.
C      *D1DP & D2DP ONLY ACCESSIBLE IN UNSTEADY FLOWS.
C+++++GROUND SERVICE SUBROUTINES:
C      *USE CONTUR(NAME,IPLANE,ILOC,NINT,I1,I2,J1,J2,GARRAY,NDIM) FOR
C        LINE-PRINTER PLOTS OF CONTOURS. 'NAME' = U1,...C4;
C        'IPLANE'= XPLANE, YPLANE, OR ZPLANE; ILOC SETS IX, IY, OR
C        IZ LOCATION OF IPLANE; I1, I2, J1, & J2 SET FIRST & LAST
C        CELLS IN HORIZ. & VERT. ON PLOT; GARRAY IS 1-D WORKING ARRAY
C        OF DIMENSION NX*NY, NX*NZ, OR NY*NZ DICTATED BY IPLANE; &
C        NDIM SETS VALUE OF DIMENSION OF GARRAY.
C      *USE FLD2DA(TITLE,GARRAY,NY,NX) TO PRINT ANY ARRAY DIMENSIONED
C        GARRAY(NY,NX); SET 'TITLE' TO REQUIRED NAME ( 4 HOLLERITH
C        CHARACTERS ONLY).
C      *USE FLD3DA(TITLE,GARRAY,NX,NY,NZ,IPLANE,ILOC) TO PRINT ANY
C        ARRAY DIMENSIONED GARRAY(NX,NY,NZ) IN PLANE SPECIFIED BY
C        'IPLANE' & 'ILOC' AS FOR CONTUR ABOVE; SET 'TITLE' AS FOR
C        FLD2DA.
C        VARIABLE NAMES FOR USE IN GROUND:
         COMMON/TYPE/CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME,WALL
         COMMON/VAR/P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,
      &KE,EP,H1,H2,H3,C1,C2,C3,C4,RX,RY,RZ,S1,S2
         COMMON/VAROLD/P10,PP0,U10,U20,V10,V20,W10,W20,R10,R20,RS0,
      &KE0,EP0,H10,H20,H30,C10,C20,C30,C40,RX0,RY0,RZ0,S10,S20
         COMMON/VARLOW/P1L,PPL,U1L,U2L,V1L,V2L,W1L,W2L,R1L,R2L,RSL,
      &KEL,EPL,H1L,H2L,H3L,C1L,C2L,C3L,C4L,RXL,RYL,RZL,S1L,S2L
         COMMON/VARHI/P1H,PPH,U1H,U2H,V1H,V2H,W1H,W2H,R1H,R2H,RSH,
      &KEH,EPH,H1H,H2H,H3H,C1H,C2H,C3H,C4H,RXH,RYH,RZH,S1H,S2H
         COMMON/GMTRY/VOL,VOL0,AEAST,ANORTH,AHIGH,AEDX,ANDY,AHDZ
         COMMON/PROP/D1,D2,D1DP,D2DP,MU1,MU1LAM,EXCO,CFP,MDT,HST1,HST2
         COMMON/PRPOLD/D10,D20
         COMMON/PRPLOW/D1L,D2L,EXCOL
         COMMON/PRPHI/D1H,D2H,MU1H,EXCOH
         COMMON/VARNX/XG,XU,DXU,DXG
         COMMON/VARNY/YG,YV,DYV,DYG,R,RV
         COMMON/VARNZ/ZG,ZW1,DZW,DZG,WGRID
         COMMON/GDMSCI/XPLANE,YPLANE,ZPLANE,ITNO
         COMMON/GDMSCL/LSLAB,MSLAB,HSLAB,LAMMU
```

```
      REAL NORTH,LOW
      INTEGER    P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,
     &EP,H1,H2,H3,C1,C2,C3,C4,RX,RY,RZ,S1,S2
      INTEGER    P10,PP0,U10,U20,V10,V20,W10,W20,R10,R20,RS0,
     &EP0,H10,H20,H30,C10,C20,C30,C40,RX0,RY0,RZ0,S10,S20
      INTEGER    P1L,PPL,U1L,U2L,V1L,V2L,W1L,W2L,R1L,R2L,RSL,
     &EPL,H1L,H2L,H3L,C1L,C2L,C3L,C4L,RXL,RYL,RZL,S1L,S2L
      INTEGER    P1H,PPH,U1H,U2H,V1H,V2H,W1H,W2H,R1H,R2H,RSH,
     &EPH,H1H,H2H,H3H,C1H,C2H,C3H,C4H,RXH,RYH,RZH,S1H,S2H
      INTEGER VOL,VOL0,AEAST,ANORTH,AHIGH,AEDX,ANDY,AHDZ
      INTEGER D1,D1DP,D2,D2DP,EXCO,CFP,HST1,HST2
      INTEGER D10,D20,D1L,D2L,EXCOL,D1H,D2H,EXCOH
      INTEGER XG,XU,DXU,DXG,YG,YV,DYV,DYG,R,RV,ZG,ZW1,DZW,
     &DZG,WGRID
      INTEGER XPLANE,YPLANE,ZPLANE
      LOGICAL LSLAB,MSLAB,HSLAB,LAMMU,LSPDA
      EQUIVALENCE (M1,R1),(M2,R2)
      SATLIT-EQUIVALENT IRUN:
      EQUIVALENCE (IRUN,INTGR(11))
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS:
      ARRAYS ( DIMENSIONED NY,NX ) FOR USE WITH 'ADD':
      DIMENSION CVAR(100,1),VVAR(100,1),CM(100,1),VM(100,1),ZERO(1,1)
      SPECIAL-DATA ARRAYS DIMENSIONED & DIMENSION VALUES SET HERE:
      DIMENSION LSPDA(1),ISPDA(1),RSPDA(1)
      USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.
      DIMENSION GH1(100,1), GC1(100,1),GC2(100,1),GC3(100,1)
      DIMENSION GH2(100,1),GD1(100,1),GD2(100,1),GR2(100,1)
      DIMENSION GTEMP(100,1),GK0(3),GK1(3),GK2(3), GK3(3), GK4(3)
      DIMENSION GM0(3),GM1(3), GM2(3),GM3(3),GM4(3)
      DIMENSION GCP0(3),GCP1(3), GCP2(3),GCP3(3), GCP4(4)
      DIMENSION GK(3), GM(3), GC(3),GX(3),GW(3),GWI(100,1)
      DIMENSION GD(4,100,1), GVOL(100,1),GV1(100,1)
      DIMENSION GN(100,1), GYP(100,1), GYNV(100,1), GAN(100,1)
      DIMENSION GXO(100,1), GXMET(100,1),GGAMH1(100,1), GMDT(100,1)
      DIMENSION GCPI(100,1), GDD(4,4),GVISC(100,1)
      DIMENSION GCP(3,100),GTEMP0(100,1)
      NAMELIST /MOLE/ATIME,GW,GWI
      NAMELIST /TEMP/ ATIME,GTEMP,GXMET
      NAMELIST /PROPM/ ATIME,GKMF,GCPMF,GROF,GFOX,GHFU,GHFGOX,GHFGM,GEW
      NAMELIST /HT/GH200,GHOX0,GCPOX,GTMOX,GROOX,GLSOX,GTOXV,GTMVA,GTMM
      NAMELIST /KSPEC/GK0,GK1,GK2,GK3,GK4
      NAMELIST /MUSPEC/GM0,GM1,GM2,GM3,GM4
      NAMELIST/DDSPEC/ GDD
      NAMELIST /CPSPEC/ GCP0,GCP1,GCP2,GCP3,GCP4
      NAMELIST /CPSPI/ ATIME,GCP,GCPI,GX
      NAMELIST /SUMKUD/ GSUMK,GSUMU,GSUMD,GSUMF
      NAMELIST/YPV/ GVOL,GAN
      NAMELIST /CINI/ ATIME,GC1,GC2,GC3
      DATA NLSP,NISP,NRSP/1,1,1/
      DATA CVAR,VVAR,CM,VM,ZERO/401*0.0/
      USER PLACES HIS DATA STATEMENTS HERE.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.
**********************************************
             GROUND  INPUT  FOR METAL COMBUSTION MODEL

**********************************************
      DATA JING,GTCR,GQOX,GBOX,GDOX,GNOX,GMOX/0,1700.,3.E4,1.,1.E-6,
     &0.5,1/
      DATA GAQ,GKA/3.E5,1.0/
      DATA GFST,GFG,GEG,GNMI,GVP,GSIG,GA0,GAI,GC1MIN/
     &   0.99843,0.111898,0.0001,4.E08,3.2E-8,5.668E-8,0.5,0.5,0.001/
      DATA GPT,GKMF,GCPMF,GROF,GFOX,GHFU,GHFGOX,GHFGM,GH1M,GH3M,GEW/
     & 100000.,121.3,1083.65,2340.,1.886532,4.45E07,4.728E06,1.13E07,
     & 0.0973E07,1.378E07,0.09/
      DATA GH200,GHOX0,GCPOX,GTMOX,GROOX,GLSOX,GTOXV,GTMM,GTMVA/
     &-1.31E07,1.643E07,1271.936,2323.,4000.,1.067E06,3253.,900.,2323./
      DATA GK0(1),GK1(1),GK2(1),GK3(1),GK4(1),GK0(2),GK1(2),GK2(2),
     &GK3(2),GK4(2),GK0(3),GK1(3),GK2(3),GK3(3),GK4(3)/
     &2133.33,26.54,0.,0.,0.,125490.,300.172,0.,0.,0.,-20915.,134.9,0.,
     &0.,0./
      DATA GM0(1),GM1(1),GM2(1),GM3(1),GM4(1),GM0(2),GM1(2),GM2(2),
```

```
     &GM3(3),GM4(2),GM0(3),GM1(3),GM2(3),GM3(3),GM4(3)/
     &33.,0.23395,0.,0.,0.,100.,0.1008857,0.,0.,0.,56.,0.2975,0.,0.,0./
      DATA GCP0(1),GCP1(1),GCP2(1),GCP3(1),GCP4(1),GCP0(2),GCP1(2),
     &GCP2(2),GCP3(2),GCP4(2),GCP0(3),GCP1(3),GCP2(3),GCP3(3),GCP4(3)/
     & 770.45,0.,0.,0.,0.,2307.5865,0.,0.,0.,0.,16497.4,0.,0.,0.,0./
      DATA GW(1),GW(2),GW(3)/ 26.98,18.016,2.01594/
      DATA GDD(1,1),GDD(1,2),GDD(1,3),GDD(2,1),GDD(2,2),GDD(2,3),
     & GDD(3,1),GDD(3,2),GDD(3,3)/
     & 0.418,1.49,0.664,1.49,3.85,4.35,0.664,4.35,0.57/
      DATA GTEMP/23*900.0,1500.0,1500.0,1800.0,3*2323.0,2500.0,
     & 2700.0,2900.0,3100.0,3500.0,3200.0, 3000.0,64*500.0/
      DATA GTEMP0/ 25*900.,10*2000.,25*900.,40*500./
      DATA GXMET/26*0.0,0.5,0.75,72*1.0/
      IF (ATIME .GT. 0.0) GOTO 2900
      PRINT PROPM
      PRINT HT
      PRINT KSPEC
      PRINT CPSPEC
      PRINT MUSPEC
      PRINT DDSPEC
      GTMJL=GTMVA
      GTMJS=GTMVA
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS:
C     PLEASE DO NOT ALTER, OR RE-SET, ANY OF THE REMAINING
C     STATEMENTS OF THIS SECTION.
 2900    IF(SPDATA)
     &CALL RDSPC(IRN,INTGR(12),LSPDA,NLSP,ISPDA,NISP,RSPDA,NRSP)
      CALL GRDUTY(IRN,ICHAP,IZED,INDVAR)
         IF(ICHAP.EQ.-5) GO TO 10
         IF(ICHAP.LE.0.OR.ICHAP.GT.16) RETURN
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,
     &1300,1400,1500,1600),ICHAP
      RETURN
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS:
C-----------------------------------------------------------------
C     CHAPTER 0: MODIFY SATLIT DATA, AT START OF EACH IRN.
C-----------------------------------------------------------------
   10 CONTINUE
C  END OF INITIAL CONDITION
C         IF(.NOT.NAMLST) RETURN
C         IF(IRN.EQ.NRUN) DATFIL=.FALSE.
C---  READ SATLIT DATA NAMELIST HERE
C     CALL WRIT40(40HENTER NAMELIST DATA FOR GROUPS 1 TO 24  )
C     READ(20,G1G24)
C     CALL WRIT40(40HENTER NAMELIST DATA FOR GROUPS 25 TO 42 )
C     READ(20,G25G42)
      RETURN
C------------------------------------------------------------------
C     CHAPTER 1: CALLED AT THE START OF EACH TIME STEP.
C     SET 'DT' HERE WHEN TLAST SET NEGATIVE IN BLOCK DATA.
C     'ATIME + DT' GIVES THE END TIME OF THE CURRENT TIME STEP.
C     NOT ACCESSED IF STEADY,OR PARABOLIC.
C------------------------------------------------------------------
  100 CONTINUE
      ATIME=ATIME+DT
      RETURN
C------------------------------------------------------------------
C     CHAPTER 2: CALLED AT THE START OF EACH SWEEP.
C------------------------------------------------------------------
  200 CONTINUE
      RETURN
C------------------------------------------------------------------
C     CHAPTER 3: CALLED AT THE START OF EACH SLAB;
C     NOT ACCESSED IF PARABOLIC, BUT 'STRIDE' IS.
C------------------------------------------------------------------
  300 CONTINUE
      RETURN
C------------------------------------------------------------------
C     CHAPTER 4: CALLED AT THE START OF EACH RE-CALCULATION OF
C     VARIABLES P1,...C4 AT CURRENT SLAB. ITNO= ITERATION NUMBER.
C------------------------------------------------------------------
```

```
00 CONTINUE
   RETURN
------------------------------------------------------------------
   CHAPTER 5: GROUND CALLED WHEN SOURCE TERM IS COMPUTED.
   INDVAR GIVES DEPENDENT VARIABLE IN QUESTION IE. U1,...C4.
   TO ADD SOURCE TO DEPENDENT VARIABLE C1(SAY) FOR IX=IXF,IXL
   AND IY=IYF,IYL INSERT STATEMENT:
   IF(INDVAR.EQ.C1)
  &CALL ADD(INDVAR,IXF,IXL,IYF,IYL,TYPE,CM,VM,CVAR,VVAR,NY,NX)
   NOTES ON 'ADD':
  *SOURCE= (CVAR(IY,IX)+AMAX1(0.0,MASFLO))*(VVAR(IY,IX)-PHI),
   WHERE 'PHI' IS IN-CELL VALUE OF VARIABLE IN QUESTION.
  *'MASFLO'= CM(IY,IX)*(VM(IY,IX)-P),
   WHERE 'P' IS THE IN-CELL PRESSURE.
  *FOR INDVAR= M1, OR =M2, SOURCE ADDED IS 'MASFLO' ONLY,
   EXCEPT FOR ONEPHS=.F. & MASFLO < 0.0 (IE. OUTFLOW) WHEN
   CM(IY,IX) IS MULTIPLIED BY R1*D1 (FOR M1) & R2*D2 (FOR M2).
  *BOTH 'CVAR' & 'CM' ARE MUTLIPLIED BY CELL-GEOMETRY QUANTITY
   DICTATED BY SETTING OF 'TYPE' (=CELL, EAST AREA,..VOLUME).
  *TYPE-SPECIFIED AREAS ARE CALCULATED AS IF BLOCKAGE ABSENT,
   BUT 'VOLUME' WITH ACCOUNT FOR ITS PRESENCE.
  *FOR ALL SOLVED VARIABLES, INCLUDING M1 ( & M2 WHEN ONEPHS=F),
   IF 'CM'> 0.0 CALL 'ADD'; FOR M1 & M2 ALTHOUGH 'CVAR' & 'VVAR'
   HAVE NO SIGNIFICANCE THEY MUST BE ENTERED AS ARGUMENTS.
  *'CVAR', 'VVAR', 'CM' & 'VM' MUST BE DIMENSIONED NY,NX.
------------------------------------------------------------------
00 CONTINUE

*************   CHAPTER     5      ******************************

**********************************************************************
COMPUTATION  OF  SMOKE OXIDE CLOUD AND METAL SURFCE LOCATION AT NEW T
**********************************************************************
   CALL GET(C1,GC1,NY,NX)
   CALL GET(C2,GC2,NY,NX)
   CALL GET(C3,GC3,NY,NX)
   CALL GET(ANORTH,GAN,NY,NX)
   CALL GET(R2,GR2,NY,NX)
   CALL GET (VOLO,GVOL,NY,NX)
   CALL GET1D(YG,GYP,NY)
   CALL GET1D(YV,GYNV,NY)
   CALL GET (MDT,GMDT,NY,NX)
   CALL GET (D1,GD1,NY,NX)
   CALL GET (D2,GD2,NY,NX)
   CALL GET(H1,GH1,NY,NX)
   CALL GET(H2,GH2,NY,NX)
   CALL GET(V1,GV1,NY,NX)
    PRINT YPV
   IF ( ATIME .GT. DT) GOTO 2700
   JL=26
   JS=28
GAS PHASE ZONE PROPERTIES: K,MU, CPI,CP,D
00 DO 2800 I=1,JL
   WRITE(6,*) '************ GTEMP=',GTEMP(I,1),I
   GT=GTEMP(I,1)
   GT2=GT**2.
   GT3=GT2*GT
   GT4=GT3*GT
   GCPI(I,1)=GCPMF
   GWI(I,1)=GW(1)
   DO 2850 IN=1,3
   GCP(IN,I)=GCP0(IN)+GCP1(IN)*GT+GCP2(IN)*GT2+GCP3(IN)*GT3
  &+GCP4(IN)*GT4
50 CONTINUE
00 CONTINUE
   DO 60 I=JL,NY
   GT=GTEMP(I,1)
   GT2=GT**2.
   GT3=GT2*GT
   GT4=GT3*GT
   IT=1
   DO 61 IN=1,3
```

```
      GK(IN)=GK0(IN)+GK1(IN)*GT+GK2(IN)*GT2+GK3(IN)*GT3+GK4(IN)*GT4
      GM(IN)=(GM0(IN)+GM1(IN)*GT+GM2(IN)*GT2+GM3(IN)*GT3+GM4(IN)*GT4)*
     &1.E-6
      GCP(IN,I)=GCP0(IN)+GCP1(IN)*GT+GCP2(IN)*GT2+GCP3(IN)*GT3
     &+GCP4(IN)*GT4
   61 CONTINUE
      GCPI(I,1)=GCP(1,I)*GC1(I,1)+GCP(2,I)*GC2(I,1)+GCP(3,I)*GC3(I,1)
      KK=1
      IF((GW(1)*GW(2)*GW(3)) .GT. 1.0E-27) GOTO 3000
      PRINT 4000,KK,I,GW(1),GW(2),GW(3)
 4000 FORMAT(2I5,3F12.3)
 3000 GWT=GC1(I,1)/GW(1) +GC2(I,1)/GW(2) + GC3(I,1)/GW(3)
      GWI(I,1)=1./GWT
      KK=2
      IF ( GWI(I,1) .GT. 1.E-27) GOTO 3100
      PRINT 4000,KK,I,GWI(I,1),GWT
 3100 GX(1)=GC1(I,1)*GW(1)/GWI(I,1)
      GX(2)=GC2(I,1)*GW(2)/GWI(I,1)
      GX(3)=GC3(I,1)*GW(3)/GWI(I,1)
      KK=3
      IF ( GTEMP(I,1) .GT. 1.0E-27 ) GOTO 3200
      PRINT 4000,KK,I,GTEMP(I,1)
 3200 GD1(I,1)=GPT*GWI(I,1)/(8483.*GTEMP(I,1))
      GD2(I,1)=GROOX
      GSUMK=0.
      GSUMU=0.
      DO 4 IN=1,3
      GSUMF=0.
      GSUMD=0.
      DO 5 IK=1,3
      KK=4
      IF( GDD(IN,IK) .GT. 1.E-27) GOTO 3300
      PRINT 4000,KK,I,GDD(IN,IK)
 3300 GSUMD=GSUMD+GX(IK)/(GDD(IN,IK)*GTEMP(I,1)**1.5)
      KK=5
      IF((GM(IK) * GW(IK)).GT. 1.0E-27) GOTO 3400
      PRINT 4000,KK,I,GM(IK),GW(IK)
 3400 GMM=GM(IN)/GM(IK)
      GWR=GW(IN)/GW(IK)
      GXNK=(1./SQRT(8*(1+GWR)))*(1+SQRT(GMM)/SQRT(SQRT(GWR)))**2.
      GSUMF=GSUMF+GMM*GXNK
    5 CONTINUE
      KK=6
      IF ( GSUMF*GSUMD*GCPI(I,1) .GT. 1.E-27) GOTO 3500
      PRINT 4000,KK,I,GSUMF,GSUMD,GCPI(I,1)
 3500 GSUMK=GSUMK+GX(IN)/GSUMF
      GSUMU=GSUMU+GX(IN)/GSUMF
      GDINI=GX(IN)/GSUMD
      GD(IN,I,1)=GD1(I,1)*GDINI
    4 CONTINUE
      GGAMH1(I,1)=GSUMK/GCPI(I,1)
      GVISC(I,1)=GSUMU
   60 CONTINUE
CD    PRINT MOLE
CD    PRINT CPSPI
CD    PRINT SUMKUD
CD    PRINT CINI
      JLL=JL-1
      DO 62 I=1,JLL
C  FLUID PHASE PROPERTIES: ROO, MU, GAMMA=K/CP
      GD1(I,1)=GROF*GCPMF/GCPI(JL+1,1)
      GD2(I,1)=1.E-10
      GVISC(I,1)= 10000.
      GD(1,I,1)=1.E-10
      GD(2,I,1)=1.E-10
      GD(3,I,1)=1.E-10
      KK=7
      IF ( GCPMF*GTMVA .GT. 1.E-27) GOTO 3600
      PRINT 4000,KK,I,GCPMF,GTMVA
 3600 GGAMH1(I,1)=GKMF/GCPMF
   62 CONTINUE
C  DENSITY AT THE SUFACE JL
```

```
.   GROG=GPT*GWI(JL,1)*GC1(JL,1)/(GTMJL*8483.)
    KK=8
    IF ( GROF* GCPOX*GROG  .GT. 1.E-27) GOTO 3700
    PRINT 4000,KK,I,GROF,GCPOX,GROG
)0 GD1(JL,1)=1./(1./GROF+GXMET(JL+1,1)*(1./GROG-1./GROF))
    GD2(JL,1)=GROOX
IECKING IF IGNITION STARTS
    IF( JIGN .NE. 0) GOTO 2750
    IF ( GTEMP(JL,1) .GE. GTCR ) GOTO 2710
;NITION YET NOT STARTED
.LCULATTION OF HEATING SOURCE AT THE METAL SURFACE
    DO 2720 IY=1,NY
    CM(IY,1)=0.
    VM(IY,1)+0.
    CVAR(IY,1)=1.E-10
    IF( IY .NE. JL ) GOTO 2720
    GDOXT=GBOX*(GC2(JL+1)**GMOX/GDOX**GNOX)*EXP(-GAQ/GTEMP(JL,1))
    VVAR(IY,1)=GROOX*GQOX*GDOXT*1.E10
    GDOX=GDOX + GDOXT*DT
 0 CONTINUE
    CALL ADD(H1,1,NX,1,NY,VOLUME,CM,VM,CVAR,VVAR,NY,NX)
    GOTO 105
MBUSTION STAGE
0 JIGN=1
   JSFLAG=0.
   JRFLAG =0.
   GSUMN=0.
   GSUMV=0.
   DO 111 I=1,NY
   IY=NY-I+1
   GT=GTEMP(IY,1)
   GTEMPO(IY,1)=(GH2(IY,1)-GHOX0-GCPOX*298)/GCPOX
   GCPDT=GCPI(IY,1)*(GTEMP(IY,1)-298)
   GCH2O=GC2(IY,1)*GH200
   GCHFG=GC1(IY,1)*(GCP(1,IY)-GCPMF)
   KK=9
   IF ( GVP*GCPMF .GT. 1.E-27 ) GOTO 3800
   PRINT 4000,KK,I,GVP,GCPMF
0 GN(IY,1)=GVOL(IY,1)*GR2(IY,1)/GVP
   IF ( GN(IY,1) .LT. GNMI ) GOTO 11
0. OF OXIDE PARTICLES BECOME SIGNIFICANT
   JCO=IY
   JRFLAG=1
   GOTO 12
1 IF (JRFLAG .NE. 1) GOTO 13
   IF (GN(IY,1) .GT. GNMI) GOTO 12
.OF OXIDE PARTICLES BECAME INSIGNIFICANT AGAIN, GOING OUT OF CLOUD
   JCI=IY
   JRFLAG=0.
   GOTO 13
2 GSUMN=GSUMN+GN(IY,1)*GVOL(IY,1)
   GSUMV=GSUMV+GVOL(IY,1)
3 IF( GC1(IY,1) .LE. GC1MIN) GOTO18
   GPAL=(GC1(IY,1)*GWI(IY,1)/GW(1))*GPT
   GTMVA= 36560.8/(25.016-ALOG(GPAL))
   GXMETA=(GH1(IY,1)-(GH1M+ GCPMF*(GTMVA-GTMM))*GC1(IY,1) -
  & (GH200+ GCP(2,IY)*(GTMVA-373.0))*GC2(IY,1) -
  & GCP(3,IY)*(GTMVA-298.0)*GC3(IY,1))/ (GHFGM*GC1(IY,1))
   IF(GXMETA .LT. 0.0) GXMETA=0.0
   IF ( IY .GT. JS ) GOTO 18
   IF ( JSFLAG .EQ. 0.) GOTO 14
   GXMETA=(GH1(IY,1)-(GH1M+ GCPMF*(GTMJS-GTMM))*GC1(IY,1) -
  & (GH200+ GCP(2,IY)*(GTMVA-373.0))*GC2(IY,1) -
  & GCP(3,IY)*(GTMVA-298.0)*GC3(IY,1))/ (GHFGM*GC1(IY,1))
   IF ( JSFLAG .GT. 1 ) GOTO 17
FLAG = 1  WHERE 0<= X =>1
   IF ( GXMETA .GT. 0.) GOTO 20
=0 FOR THE FIRST TIME
   JL=IY
   JSFLAG=2
   GXMET(IY,1)=0.0
   GTMJL=GTMVA
```

```
     GTMJS=GTMVA
     GTEMP(IY,1)=GTMJL
     GOTO 20
  17 GTEMP(IY,1)=(GH1(IY,1)-GH1M+GCPMF*GTMM)/GCPMF
      GXMET(IY,1)=0.0
     GOTO 111
 CHECKING IF SATURATION OCCOURE
  14  KK=10
      IF (GPT*GTMVA*GEW .GT. 1.E-27) GOTO 3900
      PRINT 4000,KK,GPT,GTMVA,GEW
3900  CONTINUE
     IF ( GXMETA .LE. 1 )  GOTO 19
 X>1, GAS PHASE
  18 GXMET(IY,1)=1.
     GTEMP(IY,1)=(GH1(IY,1)-(GH3M-GCP(1,IY)*GTMJS)*GC1(IY,1) -
    & (GH200-GCP(2,IY)*373.0)*GC2(IY,1) + GCP(3,IY)*298.0*GC3(IY,1)) /
    & GCPI(IY,1)
     GOTO 111
 X<=1 ,FOR THE FIRST TIME AT THIS SWEEP
  19 JS=IY
     GTMJS=GTMVA
     JSFLAG=1
  20 GCPIF=GCPI(IY,1)-(GCPI(IY,1)-GCPMF)*GC1(IY,1)
     IF ( GCPIF .GT. 1.E-10) GOTO 22
     WRITE(6,*)'***GCPIF,GCPI,GC1,IY',GCPIF,GCPI(IY,1),GC1(IY,1),IY
22   GTEMP(IY,1)=(GH1(IY,1)-(GH1M-GCPMF*GTMM+GXMETA*GHFGM)*GC1(IY,1)
    & - (GH200-GCP(2,IY)*373.0)*GC2(IY,1)+ GCP(3,IY)*GC3(IY,1)*298.0)
    & / GCPIF
     GXMET(IY,1)= GXMETA
 111 CONTINUE
     PRINT TEMP
     WRITE(6,*)'**JL= JS=',JL,JS,'***TMJL TMJS=',GTMJL,GTMJS
     GTW= GTEMP(JL,1)
     GTW4= GTW**4
     IF ( GSUMN .LE. 1.E-10 ) GOTO 2950
 RADIATION LOSES FROM THE OXIDE CLOUD
     KK=105
     IF(GSUMV .GT. 1.E-27) GOTO 5000
     PRINT 4000,KK,JCO,JCI,GSUMV
5000 GNN=GSUMN/GSUMV
     GL=GYP(JCO,1)-GYP(JCI,1)
     GRC=GYP(JCI,1)+0.5*GL
     GEP=(4.E-12)*GNN
     GRPW=GYP(JL,1)
     GEP=1.-EXP(-GEP*GL)
     GFCW=1./(1./GEW+1./GEP-1)
     JC=(JCO+JCI)/2
     GTOX=GTEMP(JC,1)
     GTOX4= GTOX**4
     JG=(JCO+NY)/2
     GTG=GTEMP(JG,1)
     GTG4=GTG**4
     GAP=GEP
     GSIG=5.6688E-8
     GAO=0.5
     GAI=0.5
     GRFC=(GAP*(GFCW*GAI*GEW*GTW4+GEG*GTG4*GAO+GEP*(1-GEW)*GFCW*GAI*
    & GTOX4)-GEP*GTOX4)*GSIG
     KK=106
     IF(GL .GT. 1.0E-27 ) GOTO 5100
     PRINT 4000,KK,JCI,GL
5100 GRCI=-GRFC/GL
2950 CONTINUE
     WRITE(6,*) '**BEGIN SOURCE,INDVAR TIME SWEEP',INDVAR,ATIME,LSWEEP
     GOTO (105,105,105,105,105,105,105,105,105,105,105,105,105,
    & 401,501,105,601,701,801,105) INDVAR
 SOURCE TERM FOR C2 : OXIDIZER
 701 DO 710 IY=2,NY
     CM(IY,1)=0.
     VM(IY,1)=0.
     CVAR(IY,1)=1.E-10
     GQ=GAQ/GTEMP(IY,1)
```

```
    IF (GQ .GE.  5. ) GQ=5.
    GKR=GKA*EXP(-GQ)
    GCDT=-3*GKR*GC2(IY,1)**3.*GC1(IY,1)**2.
    IF( GV1(IY,1) .LT. 0.0 ) GOTO 750
    GROU=(GD1(IY,1)+GD1(IY-1,1))*0.5*GV1(IY-1,1)
     GYN=GYP(IY,1)-GYP(IY-1,1)
    GROUOX=GROU*(GC2(IY,1)-GC2(IY-1,1))/GYN
            GOTO 740
50  GROU=(GD1(IY,1)+GD1(IY+1,1))*0.5*GV1(IY,1)
     GYN=GYP(IY+1,1)-GYP(IY,1)
    GROUOX=GROU*(GC2(IY+1,1)-GC2(IY,1))/GYN
i0  IF((ABS(GCDT)-ABS(GROUOX)) .GT. 1.E-5) GOTO 720
 THE REACTION IS CONTROLED BY THE RATE OF REACTION.
     VVAR(IY,1)=GCDT*1.E10
     PRINT 4000,GCDT
)00 FORMAT(5HGCDT=,F15.3)
    GMDT(IY,1)=-(1./3.)*VVAR(IY,1)*1.E-10
    GOTO 710
HE REACTION IS CONTROLED BY DIFFUSION FLUX.
'20 GF=GC1(IY,1)/GC2(IY,1)
    IF(GF .LE. GFST) GOTO 730
)IFFUSION----- FUEL RICH.
    GROUOX=-ABS(GROUOX)
    GMDT(IY,1)=-GROUOX*GFOX
    VVAR(IY,1)=GROUOX*1.E10
     PRINT 4100,GROUOX
.00 FORMAT(7HGROUOX=,F15.3)
    GOTO 710
 FUEL  LEAN
'30 VVAR(IY,1)=-ABS(GROU*(GC1(IY,1)-GC1(IY-1,1))/GYN)*1.E10
     GROUC1=VVAR(IY,1)*1.E-10
     PRINT 4200,GROUC1
200 FORMAT(7HGROUC1=,F15.3)
'10 CONTINUE
    VVAR(IY,1)=0.0
    CALL ADD(C2,1,NX,1,NY,VOLUME,CM,VM,CVAR,VVAR,NY,NX)
    WRITE(6,*)'**C2,1,25,28,35,100',GC2
    WRITE(6,*)'VV C2',VVAR
    WRITE(6,*) '******  END SOURCE ',INDVAR
    WRITE(6,*) '**ENTHALPY',GH1
    RETURN

SOURCE TERM FOR C1 : FUEL
i01 DO 610 IY=2,NY
    CM(IY,1)=0.
    VM(IY,1)=0.
    CVAR(IY,1)=1.E-10
    GQ=GAQ/GTEMP(IY,1)
    IF (GQ .GE.  5. ) GQ=5.
    GKR=GKA*EXP(-GQ)
    GCDT=-2*GKR*GC2(IY,1)**3.*GC1(IY,1)**2.
    IF( GV1(IY,1) .LT. 0.0 ) GOTO 650
    GROU=(GD1(IY,1)+GD1(IY-1,1))*0.5*GV1(IY-1,1)
     GYN=GYP(IY,1)-GYP(IY-1,1)
    GROUCM=GROU*(GC2(IY,1)-GC2(IY-1,1))/GYN
            GOTO 640
50  GROU=(GD1(IY,1)+GD1(IY+1,1))*0.5*GV1(IY,1)
     GYN=GYP(IY+1,1)-GYP(IY,1)
    GROUCM=GROU*(GC2(IY+1,1)-GC2(IY,1))/GYN
i0  IF((ABS(GCDT)-ABS(GROUCM)) .GT. 1.E-5) GOTO 620
 THE REACTION IS CONTROLED BY THE RATE OF REACTION.
     VVAR(IY,1)=GCDT*1.E10
    GMDT(IY,1)=-(1./3.)*VVAR(IY,1)*1.E-10
    GOTO 610
THE REACTION IS CONTROLED BY DIFFUSION FLUX.
520 GF=GC1(IY,1)/GC2(IY,1)
    IF(GF .LE. GFST) GOTO 630
)IFFUSION ---------   FUEL RICH.
    GROUOX=-ABS(GROU*(GC2(IY,1)-GC2(IY-1,1))/GYN)
    VVAR(IY,1)=GROUOX*GFST*1.E10
    GOTO 610
 FUEL  LEAN
```

```
  630 VVAR(IY,1)=-ABS(GROUCM)*1.E10
      GMDT(IY,1)=-GROUCM*GFOX/GFST
  610 CONTINUE
      VVAR(IY,1)=0.0
      CALL ADD(C1,1,NX,1,NY,VOLUME,CM,VM,CVAR,VVAR,NY,NX)
      WRITE(6,*)'**C1,1,25,28,35,100',GC1
      WRITE(6,*)'VV C1',VVAR
      WRITE(6,*) '******  END SOURCE ',INDVAR
      WRITE(6,*) '**ENTHALPY',GH1
      RETURN
C SOURCE TERM FOR C3 : HYDROGEN GAS PRODUCT
  801 DO 810 IY=2,NY
      CM(IY,1)=0.
      VM(IY,1)=0.
      CVAR(IY,1)=1.E-10
      GQ=GAQ/GTEMP(IY,1)
      IF (GQ .GE.  5. ) GQ=5.
      GKR=GKA*EXP(-GQ)
      GCDT=-2*GKR*GC2(IY,1)**3.*GC1(IY,1)**2.
      IF( GV1(IY,1) .LT. 0.0 ) GOTO 850
      GROU=(GD1(IY,1)+GD1(IY-1,1))*0.5*GV1(IY-1,1)
       GYN=GYP(IY,1)-GYP(IY-1,1)
      GROUCM=GROU*(GC2(IY,1)-GC2(IY-1,1))/GYN
              GOTO 840
 850  GROU=(GD1(IY,1)+GD1(IY+1,1))*0.5*GV1(IY,1)
       GYN=GYP(IY+1,1)-GYP(IY,1)
      GROUCM=GROU*(GC2(IY+1,1)-GC2(IY,1))/GYN
  840 IF((ABS(GCDT)-ABS(GROUCM)) .GT. 1.E-5) GOTO 820
C  THE REACTION IS CONTROLED BY THE RATE OF REACTION.
       VVAR(IY,1)=GCDT*1.E10
      GOTO 810
C THE REACTION IS CONTROLED BY DIFFUSION FLUX.
  820 GF=GC1(IY,1)/GC2(IY,1)
      IF(GF .LE. GFST) GOTO 830
C DIFFUSION ---------   FUEL RICH.
      GROUOX=-ABS(GROU*(GC2(IY,1)-GC2(IY-1,1))/GYN)
      VVAR(IY,1)=-GROUOX*GFG*1.E10
      GOTO 810
C  FUEL  LEAN
  830 VVAR(IY,1)=-ABS(GROUCM)*(GFG/GFST)*1.E10
  810 CONTINUE
      VVAR(IY,1)=0.0
      CALL ADD(C3,1,NX,1,NY,VOLUME,CM,VM,CVAR,VVAR,NY,NX)
      WRITE(6,*)'**C3,1,25,28,35,100',GC3
      WRITE(6,*)'VV C3',VVAR
      WRITE(6,*) '******  END SOURCE ',INDVAR
      WRITE(6,*) '**ENTHALPY',GH1
      RETURN
C HEAT SORCE TERM FOR H1
  401 DO 410 IY=1,NY
      CM(IY,1)=0.
      CVAR(IY,1)=1.E-10
      GT=GTEMP(IY,1)
      GPRP=GGAMH1(IY,1)
      GPRS=GGAMH1(IY-1,1)
      GPRN=GGAMH1(IY+1,1)
      GYNN=GYP(IY+1,1)-GYNV(IY,1)
      KK=12
      IF(GAN(IY,1) .GT. 1.E-27)GOTO 4200
      PRINT 4000,KK,IY,GAN(IY,1)
 4200 GYN=GYNV(IY,1)-GYP(IY,1)
      GYS=GYP(IY,1)-GYNV(IY-1,1)
      GYSS=GYNV(IY-1,1)-GYP(IY-1,1)
      GPRNX=GPRN*(1-GXMET(IY+1,1))
      GPRPX=GPRP*(1-GXMET(IY,1))
      GPRSX=GPRS*(1-GXMET(IY-1,1))
      GPRNC=GPRN*GC1(IY+1,1)
      GPRPC=GPRP*GC1(IY,1)
      GPRSC=GPRS*GC1(IY-1,1)
      KK=13
      IF(GPRNX*GPRPX*GPRSX .GT. 1.E-27) GOTO 4300
      PRINT 4000,KK,IY,GPRNX,GPRPX,GPRSX
```

```
0 KK=14
  IF(GPRNC*GPRPC*GYNN .GT. 1.E-27) GOTO 4400
  PRINT 4000,KK,IY,GPRNC,GPRPC,GYNN
0 KK=15
  IF(GPRN*GPRP*GCP(1,IY+1) .GT. 1.E-27) GOTO 4500
  PRINT 4000,KK,IY,GPRN,GPRP,GCP(1,IY+1)
0 KK=16
  IF(GPRS*GCP(1,IY-1)*GCP(2,IY) .GT. 1.E-27) GOTO 4600
  PRINT 4000,KK,IY,GPRS,GCP(1,IY-1),GCP(2,IY)
0 KK=17
  IF(GYS*GYSS*GCP(3,IY) .GT. 1.E-27) GOTO 4700
  PRINT 4000,KK,IY,GYS,GYSS,GCP(3,IY)
0 KK=18
  IF(GCP(3,IY-1)*GPRN*GYN .GT. 1.E-27) GOTO 4800
  PRINT 4000,KK,IY,GCP(3,IY-1),GPRN,GYN
0 IF ( IY .NE. JL ) GOTO 420
RFCE OF LIQUID METAL
W IS THE VALUE OF RADIATION ABSORBED AT THE SURFACE
  GRW=-GEW*GSIG*GTW4/GYN
  IF ( GSUMN .LE. 1.E-10 ) GOTO 460
  GRW=(GEP*GTOX4*GFCW+GEG*GTEMP((JL+JCI)/2,1)**4)*GEW/GYN+GRW
0 VVAR(IY,1)=GRW*1.E10
  GOTO 430
0 IF((IY .GT. JS) .OR. (IY .LT. JL)) GOTO 430
GION WHERE  X<1 METAL SATURETION
  GSXM1=(GC1(IY+1,1)-GC1(IY,1))/(GYNN/GPRNX+GYN/GPRPX)-(GC1(IY,1)-
 &GC1(IY-1,1))/(GYS/GPRNX+GYSS/GPRSX)
  GSXM2=(GXMET(IY+1,1)-GXMET(IY,1))/(GYNN/GPRNC+GYN/GPRPC)-
 & (GXMET(IY,1)-GXMET(IY-1,1))/(GYS/GPRPC+GYSS/GPRSC)
  GSXM=GHFGM*(GSXM2-GSXM1)
  VVAR(IY,1)=GSXM
0 IF ( IY .LT. JL ) GOTO 440
EAT  RELEASED BY METAL COMBUSTION + LATENT (V&S) HEAT  OF OXIDE IY>J
HOXV FOR HEAT RELEASED BY VAPOR PHASE  OXIDE CHANGE TO LIQUID OXIDE
HOXS FOR HEAT RELEASED BY LIQUID PHASE  OXIDE CHANGE TO SOLID OXIDE
  GHOXV=0.0
  GHOXS=0.
  IF (GTEMP(IY,1) .LE. GTOXV) GHOXV=GMDT(IY,1)*GHFGOX
  IF(GTEMPO(IY,1) .LE. GTOXV) GHOXV=GHOXV+ GROOX*GR2(IY,1)*
 &GVOL(IY,1)*GHFGOX
  IF (GTEMP(IY,1) .LE. GTMOX) GHOXS=GMDT(IY,1)*GLSOX
  IF(GTEMPO(IY,1) .LE. GTMOX) GHOXS=GHOXS+ GROOX*GR2(IY,1)*
 &GVOL(IY,1)*GLSOX
  VVAR(IY,1)=VVAR(IY,1)+GMFU*GHFU + GHOXS + GHOXV
AT TRANSFERED BY CONVECTION TO THE GAS FROM THE OXIDE PARTICLES
  GAS=3.*GR2(IY,1)/2.E-6
  VVAR(IY,1)=VVAR(IY,1)+(5.23E06)*GAS*(GTEMPO(IY,1)-GTEMP(IY,1))
RRECTION OF HEAT SOURCE DUE TO ENTHALPY CONDUTION
  GDHM=(GC1(IY+1,1)-GC1(IY,1))/(GYNN/GPRN*GCP(1,IY+1)+GYN/(GPRP*
 &GCP(1,IY)))-(GC1(IY,1)-GC1(IY-1,1))/(GYS/(GPRP*GCP(1,IY))+GYSS/(
 &GPRS*GCP(1,IY-1)))
  GDH2=(GC2(IY+1,1)-GC2(IY,1))/(GYNN/(GPRN*GCP(2,IY+1))+GYN/(GPRP
 &*GCP(2,IY)))-(GC2(IY,1)-GC2(IY-1,1))/(GYS/(GPRP*GCP(2,IY))+
 & GYSS/(GPRS*GCP(2,IY-1)))
  GDH2O=(GC3(IY+1,1)-GC3(IY,1))/(GYNN/(GPRN*GCP(3,IY+1))+GYN/(GPRP*
 &GCP(3,IY))) - (GC3(IY,1)-GC3(IY-1,1))/(GYS/(GPRP*GCP(3,IY))+GYSS/
 & (GPRS*GCP(3,IY-1)))
  GH2O= ( GC3(IY+1,1)-GC3(IY,1))/(GYNN/GPRN+GYN/GPRP) -
 & (GC3(IY,1)-GC3(IY-1,1))/(GYS/GPRP+GYSS/GPRS)
  VVAR(IY,1)=VVAR(IY,1)+(GDHM+ GDH2 +GDH2O)*(GT-298)+ GH2OO*GH2O
  GOTO 450
AT SOURCE IN METAL . NO HEAT GNERATEDAT THIS RIGION IY<JL.
0 VVAR(IY,1)=0.0
0 VVAR(IY,1)=VVAR(IY,1)*1.E10
0 CONTINUE
    CALL ADD(H1,1,NX,1,NY,VOLUME,CM,VM,CVAR,VVAR,NY,NX)
  WRITE(6,*)'***H1',GH1
  WRITE(6,*)'**TEMP',GTEMP
  WRITE(6,*)'** H1 SOURE',VVAR
  WRITE(6,*) '******  END SOURCE ',INDVAR
  RETURN
AT SOURCES ON  THE OXIDE PHASE H2
```

```
  501 DO 510 IY=1,NY
      KK=19
      IF( GCPOX*GCPI(IY,1)*GVOL(IY,1) .GT. 1.E-27) GOTO 4900
      PRINT 4000,KK,IY,GCPOX,GCPI(IY,1),GVOL(IY,1)
 4900 CM(IY,1)=0.
      VM(IY,1)=0.
      GHOXS=0.0
      CVAR(IY,1)=1.E-10
      VVAR(IY,1)=0.
  510 CONTINUE
      DO 520 IY=JL,NY
      GAS=3.*GR2(IY,1)/2.E-6
C HEAT LOSES BY CONVECTION TO THE GAS PHASE
      VVAR(IY,1)=-(5.23E6)*GAS*(GH2(IY,1)*GCP(IY,1)-GH1(IY,1)*GCPOX+
     & GCPOX*(GC3(IY,1)*GH200)-GCPI(IY,1)*GHOXO)/(GCPOX*GCPI(IY,1))
      IF(((GTEMPO(IY,1)-GTMOX) .LT. 3) .OR. (GTEMP(IY,1) .LE. GTMOX))
     & GHOXS=GROOX*GR2(IY,1)*GVOL(IY,1)*GLSOX
C SOLDIFICATION OF OXIDE - HEAT LOSES TO GAS PHASE
      VVAR(IY,1)=VVAR(IY,1)-GHOXS
C RADIATION HEAT LOSES
      IF(( IY .GE.JCI) .AND. ( IY .LE. JCO)) VVAR(IY,1)=VVAR(IY,1)-GRCI
C CONTRIBUTION OF HEAT GENERATED BY NEW OXIDE COMBUSTION PRODUCT
      VVAR(IY,1)=VVAR(IY,1)+GMDT(IY,1)*GCPOX*(GTOXV-GTEMPO(IY,1))/
     &GVOL(IY,1)
      VVAR(IY,1)=VVAR(IY,1)*1.E10
  520 CONTINUE
      CALL ADD(H2,1,NX,1,NY,VOLUME,CM,VM,CVAR,VVAR,NY,NX)
      WRITE(6,*)'***H2',GH2
      WRITE(6,*)'**TEMP OX',GTEMPO
      WRITE(6,*)'** H2 SOURE',VVAR
      WRITE(6,*) '******  END SOURCE ',INDVAR
  105 RETURN
C**************END  OF CHAPTER 5   ***********************************
C--------------------------------------------------------------------
C     CHAPTER 6: CALLED AT THE END OF EACH VARIABLE-RECALCULATION
C     CYCLE COMMENCED AT CHAPTER 4. ITNO = ITERATION NUMBER.
C--------------------------------------------------------------------
  600 CONTINUE
      RETURN
C--------------------------------------------------------------------
C     CHAPTER 7: CALLED AT END OF EACH SLAB-WISE CALCULATION.
C--------------------------------------------------------------------
  700 CONTINUE
C**************  CHAPER   7     **************************************
      CALL SET(U2,1,NX,1,NY,GTEMP,NY,NX)
      CALL SET(U1,1,NX,1,NY,GTEMPO,NY,NX)
      DO 106 I=1,NY
      IF(MSLAB) GOTO 107
      JC1=C1H
      JC2=C2H
      JC3=C3H
      GOTO 108
 107  JC1=C1
      JC3=C3
      JC2=C2
  108 CALL GET(JC1,GC1,NY,NX)
      CALL GET(JC2,GC2,NY,NX)
      CALL GET(JC3,GC3,NY,NX)
      IF( I .GT.JL) GOTO 109
      GC1(I,1)=1.0
      GC2(I,1)=0.0
      GC3(I,1)=0.0
      GOTO 106
  109 IF (GC1(I,1) .GT.1.0)GC1(I,1)=1.0
      IF (GC2(I,1) .GT.1.0)GC2(I,1)=1.0
      IF (GC3(I,1) .GT.1.0)GC3(I,1)=1.0
      IF (GC1(I,1) .LT. 0.0)GC1(I,1)=0.0
      IF (GC2(I,1) .LT. 0.0)GC2(I,1)=0.0
      IF (GC3(I,1) .LT. 0.0)GC3(I,1)=0.0
 106  CONTINUE
      CALL SET(JC1,1,NX,1,NY,GC1,NY,NX)
      CALL SET(JC2,1,NX,1,NY,GC2,NY,NX)
```

```
   CALL SET(JC3,1,NX,1,NY,GC3,NY,NX)
   RETURN
-----------------------------------------------------------------
   CHAPTER 8: CALLED AT THE END OF EACH SWEEP;
   NOT ACCESSED IF PARABOLIC.
-----------------------------------------------------------------
00 CONTINUE
   RETURN
-----------------------------------------------------------------
   CHAPTER 9: CALLED AT THE END OF EACH TIME STEP;
   NOT ACCESSED IF PARABOLIC.
-----------------------------------------------------------------
00 CONTINUE
   RETURN
-----------------------------------------------------------------
   CHAPTER 10: SET PHASE 1 DENSITY HERE WHEN IRHO1=-1 IN DATA.
   SET CURRENT-Z 'SLAB' DENSITY, D1, IF MSLAB=.T.,
   EG.  IF(MSLAB) CALL SET(D1,1,NX,1,NY,GD1,NY,NX).
   SET NEXT LARGER-Z 'SLAB' DENSITY, D1H, IF HSLAB=.T. & PARAB=F
   EG.  IF(HSLAB) CALL SET(D1H,1,NX,1,NY,GD1H,NY,NX).
   SET D(LN(D1))/DP (IE. D1DP) FOR UNSTEADY FLOW,
   EG.  IF(MSLAB) CALL SET(D1DP,1,NX,1,NY,GD1DP,NY,NX).
-----------------------------------------------------------------
00 CONTINUE
*************CHAPTER  10     ****************************************
   IF (MSLAB) GOTO 101
   JD1=D1H
   GOTO 102
01 JD1=D1
02 CALL SET(JD1,1,NX,1,NY,GD1,NY,NX)
   RETURN
-----------------------------------------------------------------
   CHAPTER 11: SET PHASE 2 DENSITY HERE WHEN IRHO2=-1 IN DATA.
   SET CURRENT-Z 'SLAB' DENSITY, D2, IF MSLAB=.T.,
   EG.  IF(MSLAB) CALL SET(D2,1,NX,1,NY,GD2,NY,NX).
   SET NEXT LARGER-Z 'SLAB' DENSITY, D2H, IF HSLAB=.T. & PARAB=F
   EG.  IF(HSLAB) CALL SET(D2H,1,NX,1,NY,GD2H,NY,NX).
   SET D(LN(D2))/DP FOR UNSTEADY FLOW,
   EG.  IF(MSLAB) CALL SET(D2DP,1,NX,1,NY,GD2DP,NY,NX).
-----------------------------------------------------------------
00 CONTINUE
*********** CHAPTER 11      **********************************
   IF (MSLAB) GOTO 103
   JD2=D2H
   GOTO 104
03 JD2=D2
04 CALL SET(JD2,1,NX,1,NY,GD2,NY,NX)
   RETURN
-----------------------------------------------------------------
   CHAPTER 12: SET PHASE 1 VISCOSITY HERE WHEN IEMU1=-1 IN DATA.
   SET CURRENT-Z 'SLAB' VISCOSITY (MU1), IF MSLAB=.T.,
   EG.  IF(MSLAB) CALL SET(MU1,1,NX,1,NY,GVISC,NY,NX).
   SET NEXT LARGER-Z 'SLAB' VISC. (MU1H), IF HSLAB=.T. & PARAB=F
   EG.  IF(HSLAB) CALL SET(MU1H,1,NX,1,NY,GVSCH,NY,NX).

   CHAPTER ALSO ACCESSED WHEN EMULAM=-1.0 IN DATA, SO THAT THE
   LAMINAR VISCOSITY WHICH APPEARS IN WALL FUNCTIONS & IN THE
   KE-EP TURBULENCE MODEL (IEMU1=2) MAY BE SET NON-CONSTANT.
   SET CURRENT-Z 'SLAB' VALUE (MU1LAM) WHEN LAMMU=.T.,
   EG.  IF(LAMMU) CALL SET(MU1LAM,1,NX,1,NY,GVSCL,NY,NX).
-----------------------------------------------------------------
00 CONTINUE
***********  CHAOTER 12  ***********************************
   IF(MSLAB) CALL SET(MU1,1,NX,1,NY,GVISC,NY,NX)
   IF (HSLAB) CALL SET(MU1H,1,NX,1,NY,GVISC,NY,NX)
   IF (LAMMU) CALL SET( MU1LAM,1,NX,1,NY,GVISC,NY,NX)
   RETURN
-----------------------------------------------------------------
   CHAPTER 13: SET EXCHANGE COEFFICIENT (E.C.) FOR VARIABLE
   INDVAR WHEN SIGMA(INDVAR)=-1.0 IN DATA.
   SET CURRENT-Z 'SLAB' E.C. (EXCO) IF MSLAB=.T.,
   EG.  IF(MSLAB) CALL SET(EXCO,1,NX,1,NY,GEXCO,NY,NX).
```

```
C      SET NEXT SMALLER-Z 'SLAB' E.C. (EXCOL) IF LSLAB=.T.,
C      EG.  IF(LSLAB) CALL SET(EXCOL,1,NX,1,NY,GEXCOL,NY,NX).
C      SET NEXT LARGER-Z  'SLAB' E.C. (EXCOH) IF HSLAB=.T.,
C      EG.  IF(HSLAB) CALL SET(EXCOH,1,NX,1,NY,GEXCOH,NY,NX).
C      NOTE: FOR MSLAB, INDVAR=U1,..C4; FOR LSLAB, INDVAR=U1L,..C4L
C      & FOR HSLAB, INDVAR=U1H,..C4H. IF PARAB=.T. SET MSLAB ONLY.
C-----------------------------------------------------------------------
 1300 CONTINUE
C ************   CHAPTER 13     ***********************************
      IF( INDVAR .NE.H1) GOTO 131
      IF (LSLAB) CALL SET(EXCOL,1,NX,1,NY,GGAMH1,NY,NX)
      IF (MSLAB) CALL SET(EXCO,1,NX,1,NY,GGAMH1,NY,NX)
      IF (HSLAB) CALL SET(EXCOH,1,NX,1,NY,GGAMH1,NY,NX)
  131 IF (INDVAR .NE.C1) GOTO 132
      IF (LSLAB) CALL SET(EXCOL,1,NX,1,NY,GD(1,NY,1),NY,NX)
      IF (MSLAB) CALL SET(EXCO,1,NX,1,NY,GD(1,NY,1),NY,NX)
      IF (HSLAB) CALL SET(EXCOH,1,NX,1,NY,GD(1,NY,1),NY,NX)
  132 IF( INDVAR .NE. C2) GOTO 133
      IF (LSLAB) CALL SET(EXCOL,1,NX,1,NY,GD(2,NY,1),NY,NX)
      IF (MSLAB) CALL SET(EXCO,1,NX,1,NY,GD(2,NY,1),NY,NX)
      IF (HSLAB) CALL SET(EXCOH,1,NX,1,NY,GD(2,NY,1),NY,NX)
  133 IF (INDVAR .NE.C3 ) GOTO 134
      IF (LSLAB) CALL SET(EXCOL,1,NX,1,NY,GD(3,NY,1),NY,NX)
      IF (MSLAB) CALL SET(EXCO,1,NX,1,NY,GD(3,NY,1),NY,NX)
      IF (HSLAB) CALL SET(EXCOH,1,NX,1,NY,GD(3,NY,1),NY,NX)
  134 RETURN
C-----------------------------------------------------------------------
C      CHAPTER 14: SET INTER-PHASE FRICTION COEFFICIENT (CFP) HERE
C      WHEN ICFIP = -1 IN DATA; ITS UNITS = FORCE / (CELL * RELATIVE
C      SPEED OF PHASES).
C-----------------------------------------------------------------------
 1400 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C      CHAPTER 15: SET INTER-PHASE MASS-TRANSFER RATE PER CELL (MDT)
C      HERE WHEN IMDOT = -1 IN DATA.
C-----------------------------------------------------------------------
 1500 CONTINUE
C ************  CHAPTER  15    *************************************
      CALL SET(MDT,1,NX,1,NY,GMDT,NY,NX)
      RETURN
C-----------------------------------------------------------------------
C      CHAPTER 16: SET HERE PHASE 1 & 2 SATURATION ENTHALPIES
C      ( HST1 & HST2) WHEN IHSAT = -1 IN DATA.
C-----------------------------------------------------------------------
 1600 CONTINUE
      RETURN
      END
C$DIRECTIVE**STRIDE
      SUBROUTINE STRIDE(IZSTEP,IGOTO,IRN)
C-----------------------------------------------------------------------
C      USE THIS SUBROUTINE TO SPECIFY THE GEOMETRY
C      OF THE FORWARD STEP IN PARABOLIC CALCULATIONS.
C      IZSTEP IS THE CURRENT FORWARD STEP, & NZSTP IS THE LAST
C      FORWARD STEP (FOR PARAB=.T. EARTH SETS NZ=1 ).
C      THE COMMON VARIABLE 'ZWL' GIVES THE DISTANCE OF THE
C      PREVIOUS STEP FROM THE ORIGIN.
C-----------------------------------------------------------------------
C$INCLUDE 9,CMNGUSSI.FTN/G
      LOGICAL LOGIC1,LOGIC
      DIMENSION LOGIC(100)
      COMMON/LDATA/LOGIC1(309)
      EQUIVALENCE (LOGIC(1),LOGIC1(210))
      DIMENSION INTGR(100)
      COMMON/IDATA/INTGR1(194)
      EQUIVALENCE (INTGR(1),INTGR1(95))
      DIMENSION RE(100)
      COMMON/RDATA/RE1(421)
      EQUIVALENCE (RE(1),RE1(322))
      COMMON/BOUND/LOCREG(60),
     &TR1,CP1R1(7),VP1R1(7),CP2R1(5),VP2R1(5),CPNR1(5),VPNR1(5),
     &TR2,CP1R2(7),VP1R2(7),CP2R2(5),VP2R2(5),CPNR2(5),VPNR2(5),
```

```
  &TR3,CP1R3(7),VP1R3(7),CP2R3(5),VP2R3(5),CPNR3(5),VPNR3(5),
  &TR4,CP1R4(7),VP1R4(7),CP2R4(5),VP2R4(5),CPNR4(5),VPNR4(5),
  &TR5,CP1R5(7),VP1R5(7),CP2R5(5),VP2R5(5),CPNR5(5),VPNR5(5),
  &TR6,CP1R6(7),VP1R6(7),CP2R6(5),VP2R6(5),CPNR6(5),VPNR6(5),
  &TR7,CP1R7(7),VP1R7(7),CP2R7(5),VP2R7(5),CPNR7(5),VPNR7(5),
  &TR8,CP1R8(7),VP1R8(7),CP2R8(5),VP2R8(5),CPNR8(5),VPNR8(5),
  &TR9,CP1R9(7),VP1R9(7),CP2R9(5),VP2R9(5),CPNR9(5),VPNR9(5),
  &TR10,CP1R10(7),VP1R10(7),CP2R10(5),VP2R10(5),CPNR10(5),VPNR10(5)
    DIMENSION TCVREG(350)
    EQUIVALENCE (TCVREG(1),TR1)
NCLUDE 9,GUSSEQUI.FTN/G
    LOGICAL CARTES,POLAR,SPDATA,SKEW,TWODYZ,ONEDZ,STOVAR(25),
  &    SOLVAR(25),PRINT(25),RESID(25),CMPRSS,CONEMU,LSP1,
  &    CONRHO,EMDOT,ONEPHS,INCORE(10),ISAVED,SAVEI,SAVEM,
  &    RESTRT,XCYCLE,MONITR,REGION(10),STEADY,WHOLEP,SLABPP,
  &    RAIN,BLOCKZ,PWSTAG,RUN(30),PLOT,RESMAP,FLAG,BLOCK,
  &    TEST,CONC1(4),DISTIL
    LOGICAL SPRESS,PARAB,DONACC,OVERLY,SACC,
  &    GUSSIE,CATHY,CONNIE,CORA,ESTER,FLASH,FLORA,
  &    FOCS,GENMIX,HESTER,PICALO,PLANT,SPLASH,HELGA,
  &    TACT,TIBALT,TOPSI,PAMELA,TABLES,WSTAG,
  &    CONMOD,GROSTA,SUBPST,SUBWGR
    EQUIVALENCE (CARTES,LOGIC1(1)),(POLAR,LOGIC1(2)),
  &    (SPDATA,LOGIC1(3)),(SKEW,LOGIC1(4)),(TWODYZ,LOGIC1(5)),
  &    (ONEDZ,LOGIC1(6)),(STOVAR(1),LOGIC1(7)),
  &    (SOLVAR(1),LOGIC1(32)),(PRINT(1),LOGIC1(57)),
  &    (RESID(1),LOGIC1(82)),(CMPRSS,LOGIC1(107)),
  &    (CONEMU,LOGIC1(108)),(LSP1,LOGIC1(109)),
  &    (CONRHO,LOGIC1(110)),(EMDOT,LOGIC1(111)),
  &    (ONEPHS,LOGIC1(112)),(INCORE(1),LOGIC1(113))
    EQUIVALENCE(ISAVED,LOGIC1(123)),(SAVEI,LOGIC1(124)),
  &    (SAVEM,LOGIC1(125)),(RESTRT,LOGIC1(126)),
  &    (XCYCLE,LOGIC1(127)),(MONITR,LOGIC1(128)),
  &    (REGION(1),LOGIC1(129)),(STEADY,LOGIC1(139)),
  &    (WHOLEP,LOGIC1(140)),(SLABPP,LOGIC1(141)),
  &    (RAIN,LOGIC1(142)),(BLOCKZ,LOGIC1(143)),
  &    (PWSTAG,LOGIC1(144)),(RUN(1),LOGIC1(145)),
  &    (PLOT,LOGIC1(175)),(RESMAP,LOGIC1(176)),
  &    (FLAG,LOGIC1(177)),(BLOCK,LOGIC1(178)),(TEST,LOGIC1(179))
    EQUIVALENCE(CONC1(1),LOGIC1(206))
    EQUIVALENCE (SPRESS,LOGIC(100)),(PARAB,LOGIC(99)),
  &    (DONACC,LOGIC(98))
    EQUIVALENCE (OVERLY,LOGIC(96)),(SACC,LOGIC(95)),
  &    (WSTAG,LOGIC(94))
    EQUIVALENCE (DISTIL,LOGIC(90))
    EQUIVALENCE (PAMELA,LOGIC(67)),(TOPSI,LOGIC(66)),
  &    (TIBALT,LOGIC(65)),(TACT,LOGIC(64)),(HELGA,LOGIC(63)),
  &    (SPLASH,LOGIC(62)),(PLANT,LOGIC(61)),(PICALO,LOGIC(60)),
  &    (HESTER,LOGIC(59)),(GENMIX,LOGIC(58)),(FOCS,LOGIC(57)),
  &    (FLORA,LOGIC(56)),(FLASH,LOGIC(55)),(ESTER,LOGIC(54)),
  &    (CORA,LOGIC(53)),(CONNIE,LOGIC(52)),(CATHY,LOGIC(51)),
  &    (GUSSIE,LOGIC(50))
    EQUIVALENCE (TABLES,LOGIC(48)),(CONMOD,LOGIC(47)),
  &    (GROSTA,LOGIC(46)),(SUBPST,LOGIC(45))
    EQUIVALENCE (SUBWGR,LOGIC(43))
    INTEGER FSTEP,FSWEEP,TSTSWP,ITAB(8),MTABVR(8)
    DIMENSION ISPCSO(25),LITER(25)
    EQUIVALENCE (NX,INTGR1(1)),(NY,INTGR1(2)),
  &    (NZ,INTGR1(3)),(ISPCSO(1),INTGR1(4)),
  &    (NREGN,INTGR1(29)),(NPHI,INTGR1(30)),
  &    (LITKE,INTGR1(31)),(LITHYD,INTGR1(32)),
  &    (LITH,INTGR1(33)),(LITCNC,INTGR1(34)),
  &    (LITSLB,INTGR1(35)),(NRUN,INTGR1(36)),
  &    (LITER(1),INTGR1(37)),(FSTEP,INTGR1(62)),
  &    (FSWEEP,INTGR1(63)),(LSTEP,INTGR1(64))
    EQUIVALENCE (LSWEEP,INTGR1(65)),(NPRINT,INTGR1(66)),
  &    (IERRP,INTGR1(67)),(IMAXP,INTGR1(68)),
  &    (IEMU1,INTGR1(69)),(IXMON,INTGR1(70)),
  &    (IYMON,INTGR1(71)),(IZMON,INTGR1(72)),
  &    (KEMU,INTGR1(73)),(KMAIN,INTGR1(74)),
  &    (KINDEX,INTGR1(75)),(KGEOM,INTGR1(76)),
  &    (KINPUT,INTGR1(77)),(KSODAT,INTGR1(78))
```

```
      EQUIVALENCE(KFLAG,INTGR1(79)),(KCOMPF,INTGR1(80)),                MEL12050
     &   (KSORCE,INTGR1(81)),(KLES1D,INTGR1(82)),                       MEL12060
     &   (KLES2D,INTGR1(83)),(KLES3D,INTGR1(84)),                       MEL12070
     &   (KCOMPP,INTGR1(85)),(KADJST,INTGR1(86)),                       MEL12080
     &   (KFLUX,INTGR1(87)),(KSHIFT,INTGR1(88)),                        MEL12090
     &   (KOUTPT,INTGR1(89)),(KDIF,INTGR1(90)),                         MEL12100
     &   (KCOMPU,INTGR1(91)),(KCOMPV,INTGR1(92)),                       MEL12110
     &   (KCOMPW,INTGR1(93)),(KCOMPR,INTGR1(94))                        MEL12120
      EQUIVALENCE (KELIN,INTGR(100)),(MEANDF,INTGR(99)),                MEL12130
     &   (NUMCLS,INTGR(98)),                                            MEL12140
     &   (IRHO1,INTGR(97)),(IRHO2,INTGR(96)),                           MEL12150
     &   (IZW1,INTGR(95)),(IZW2,INTGR(94)),                             MEL12160
     &   (MGRID,INTGR(93)),(KWALL,INTGR(92)),                           MEL12170
     &   (IZPR1,INTGR(91)),(IZPR2,INTGR(90)),                           MEL12180
     &   (ISTPR1,INTGR(89)),(ISTPR2,INTGR(88)),                         MEL12190
     &   (NTPRIN,INTGR(87))                                             MEL12200
      EQUIVALENCE (IMDOT,INTGR(86)),(IHSAT,INTGR(85)),                  MEL12210
     &   (ICFIP,INTGR(84)),(NTABLE,INTGR(83)),(NTABVR,INTGR(82)),       MEL12220
     &   (LINTAB,INTGR(81)),(NPRTAB,INTGR(80)),(NMON,INTGR(79)),        MEL12230
     &   (ITAB(1),INTGR(71)),(MTABVR(1),INTGR(63)),                     MEL12240
     &   (NZPRIN,INTGR(62)),                                            MEL12250
     &   (NXPRIN,INTGR(61)),(NYPRIN,INTGR(60)),                         MEL12260
     &   (LDISTL,INTGR(59)),(TSTSWP,INTGR(58)),                         MEL12270
     &   (KDBEXP,INTGR(57)),(KDBRHO,INTGR(56)),                         MEL12280
     &   (KDBMDT,INTGR(55))                                             MEL12290
      EQUIVALENCE (ILOOP1,INTGR(53)),(ILOOPN,INTGR(52)),                MEL12300
     &   (IPBP,INTGR(51)),(IZPRF,INTGR(50)),                            MEL12310
     &   (IZPRL,INTGR(49)),(ISTPRF,INTGR(48)),                          MEL12320
     &   (ISTPRL,INTGR(47)),(KDBGEN,INTGR(46)),                         MEL12330
     &   (IVELF,INTGR(45)),(IVELL,INTGR(44)),                           MEL12340
     &   (IKEF,INTGR(43)),(IKEL,INTGR(42)),                             MEL12350
     &   (IENTF,INTGR(41)),(IENTL,INTGR(40)),                           MEL12360
     &   (ICNCF,INTGR(39)),(ICNCL,INTGR(38)),                           MEL12370
     &   (NVEL,INTGR(37)),(NKE,INTGR(36)),                              MEL12380
     &   (NENT,INTGR(35)),(NCNC,INTGR(34)),                             MEL12390
     &   (NZSTP,INTGR(33)),(NPRMNT,INTGR(32))                           MEL12400
      DIMENSION SPARE1(20),XUDIST(30),YVDIST(30),ZWDIST(50),            MEL12410
     &   SIGMA(25),CRIT(25),DTFALS(25),RESREF(25),                      MEL12420
     &   TITLE(25),FIINIT(25),TFRAC(30)                                 MEL12430
      EQUIVALENCE (SPARE1(1),RE1(1)),                                   MEL12440
     &   (TFRAC(1),RE1(21)),(XULAST,RE1(51)),                           MEL12450
     &   (YVLAST,RE1(52)),(ZWLAST,RE1(53)),                             MEL12460
     &   (XUDIST(1),RE1(54)),(YVDIST(1),RE1(84)),                       MEL12470
     &   (ZWDIST(1),RE1(114)),(SIGMA(1),RE1(164)),                      MEL12480
     &   (CRIT(1),RE1(189)),(DTFALS(1),RE1(214)),                       MEL12490
     &   (RESREF(1),RE1(239)),(EMU1,RE1(264)),                          MEL12500
     &   (RHO1,RE1(265)),(RHO2,RE1(266)),                               MEL12510
     &   (TLAST,RE1(267)),(CFIPS,RE1(268))                              MEL12520
      EQUIVALENCE (AMDOT,RE1(269)),(FIINIT(1),RE1(270)),                MEL12530
     &   (RELAXP,RE1(295)),(TITLE(1),RE1(296)),                         MEL12540
     &   (DT,RE1(321)),                                                 MEL12550
     &   (RINNER,ZWDIST(50)),(SNALFA,ZWDIST(49)),                       MEL12560
     &   (PBAR,ZWDIST(48))                                              MEL12570
      EQUIVALENCE (SLOEMU,RE(100)),(SLORHO,RE(99)),(RLXRHO,RE(98)),     MEL12580
     &   (RHOMAX,RE(97)),(RHOMIN,RE(96)),                               MEL12590
     &   (EMUMAX,RE(95)),(EMUMIN,RE(94)),                               MEL12600
     &   (TKEMAX,RE(93)),(TKEMIN,RE(92)),                               MEL12610
     &   (RLXPZ,RE(91)),                                                MEL12620
     &   (ABUOY,RE(90)),(HREF,RE(89)),                                  MEL12630
     &   (AGRAVX,RE(88)),(AGRAVY,RE(87)),                               MEL12640
     &   (AGRAVZ,RE(86)),(ATIME,RE(85))                                 MEL12650
      EQUIVALENCE (RLXPXY,RE(84)),                                      MEL12660
     &   (EPSMAX,RE(83)),(EPSMIN,RE(82)),                               MEL12670
     &   (ARHO1,RE(81)),(BRHO1,RE(80)),(CRHO1,RE(79))                   MEL12680
      EQUIVALENCE (AZW2,RE(73)),(BZW2,RE(74)),                          MEL12690
     &   (PINT,RE(75)),(CZW2,RE(76)),                                   MEL12700
     &   (DZ,RE(78)),(ZWL,RE(77)),(ZW2M1T,RE(72)),                      MEL12710
     &   (VELMIN,RE(70)),(VELMAX,RE(69)),                               MEL12720
     &   (FALCOM,RE(68)),(CMDOT,RE(67)),                                MEL12730
     &   (RLXMDT,RE(66)),(AMDTMX,RE(65)),(AMDTMN,RE(64)),               MEL12740
     &   (RADMAX,RE(63)),(RADMIN,RE(62)),                               MEL12750
     &   (PRESS,RE(61)),(CP1,RE(60)),                                   MEL12760
```

```
     &   (CP2,RE(59)),(CP3,RE(58)),(FSTOIC,RE(57)),
     &   (ARRCON,RE(56)),(PREEXP,RE(55)),(CA1I,RE(54))
      EQUIVALENCE (CA2I,RE(53)),(CA13,RE(52)),(CA23,RE(51)),
     &   (H1SAT,RE(50)),(H2SAT,RE(49)),
     &   (EMULAM,RE(48))
      EQUIVALENCE (TWPRCN,RE(42)),(PSATEX,RE(41))
      EQUIVALENCE(ELEXP,RE(39)),(EWALL,RE(38)),
     &   (SIGE,RE(37)),(SIGK,RE(36)),(TAUDK,RE(35))
      DIMENSION LOC1(6),LOC2(6),LOC3(6),LOC4(6),LOC5(6),
     &   LOC6(6),LOC7(6),LOC8(6),LOC9(6),LOC10(6)
      EQUIVALENCE (LOCREG(1),LOC1(1)),(LOCREG(7),LOC2(1))
      EQUIVALENCE (LOCREG(13),LOC3(1)),(LOCREG(19),LOC4(1))
      EQUIVALENCE (LOCREG(25),LOC5(1)),(LOCREG(31),LOC6(1))
      EQUIVALENCE (LOCREG(37),LOC7(1)),(LOCREG(43),LOC8(1))
      EQUIVALENCE (LOCREG(49),LOC9(1)),(LOCREG(55),LOC10(1))
      DIMENSION XUFRAC(30),YVFRAC(30),ZWFRAC(30),ENDIT(25)
      DIMENSION XFRAC(30),YFRAC(30),ZFRAC(30)
      EQUIVALENCE(XUDIST(1),XUFRAC(1),XFRAC(1)),
     &            (YVDIST(1),YVFRAC(1),YFRAC(1)),
     &             (ZWDIST(1),ZWFRAC(1),ZFRAC(1)),(LITCNC,LITC),
     &             (ENDIT(1),CRIT(1)),(RLXP,RELAXP)
      EQUIVALENCE (NPRMON,LITER(1))
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 STARTS:
      SATLIT-EQUIVALENT IRUN:
      EQUIVALENCE (IRUN,INTGR(11))
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS:
      USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.
      USER PLACES HIS DATA STATEMENTS HERE.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS:
      GO TO (10,20,30,40,50,60),IGOTO
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS:
-----------------------------------------------------------------
      SECTION 1: SET FORWARD STEP SIZE DZ FOR IZSTEP>1 WHEN
      ZWLAST < 0. IN DATA.
      AT IZSTEP=1 EARTH SETS DZ = ABS(ZWDIST(1)*ZWLAST)
-----------------------------------------------------------------
   10 CONTINUE
***USER SETS DZ HERE...
      RETURN
-----------------------------------------------------------------
      SECTION 2: SET X-WIDTH (XULAST) OF GRID FOR IZSTEP > 1
      WHEN XULAST < 0. IN DATA.
      AT IZSTEP=1 EARTH SETS XULAST = ABS(XULAST)
-----------------------------------------------------------------
   20 CONTINUE
***USER SETS XULAST HERE...
      RETURN
-----------------------------------------------------------------
      SECTION 3: SET Y-WIDTH (YVLAST) OF GRID FOR IZSTEP > 1
      WHEN YVLAST < 0 IN DATA.
      AT IZSTEP=1 EARTH SETS YVLAST = ABS(YVLAST)
-----------------------------------------------------------------
   30 CONTINUE
***USER SETS YVLAST HERE...
      RETURN
-----------------------------------------------------------------
      SECTION 4: SET INNER RADIUS (RINNER) OF GRID FOR IZSTEP > 1
      WHEN RINNER < 0 IN DATA.
      AT IZSTEP=1 EARTH SETS RINNER = ABS(RINNER)
-----------------------------------------------------------------
   40 CONTINUE
***USER SETS RINNER HERE...
      RETURN
-----------------------------------------------------------------
      SECTION 5: SET SLOPE (SNALFA) OF INNER EDGE OF GRID FOR
      IZSTEP > 1  WHEN SNALFA> 1, & CARTES = .FALSE.
      AT IZSTEP=1 EARTH SETS SNALFA = SNALFA-2.
-----------------------------------------------------------------
   50 CONTINUE
```

```
C*****USER SETS SNALFA HERE...
      RETURN
C-----------------------------------------------------------------------
C     SECTION 6: SET MEAN PRESSURE (PBAR) AT NEXT FORWARD STEP
C     WHEN PBAR < 0. IN DATA.
C     FOR UNCONFINED FLOWS WITH IMPRESSED NON-ZERO
C     PRESSURE GRADIENTS SET PBAR HERE; FOR CONFINED
C     FLOWS EARTH AUTOMATICALLY COMPUTES PRESSURE REQUIRED.
C-----------------------------------------------------------------------
   60 CONTINUE
C*****USER SETS PBAR HERE...
      RETURN
      END
```

```
IRECTIVE**SATLIT
  *FILE NAME: MODSTL.FTN
  *ABSTRACT: SATELLITE MODEL MAIN PROGRAM.
  *DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983).
  *AUXILIARY SUBROUTINES (TAPES, ETC.) ARE IN SATELLITE LIBRARY
   SERVICEU, WHICH MUST BE INCLUDED IN LINK EDIT TO RUN.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 STARTS:
--------------------------------------------------------------
PTER 1   COMMON BLOCKS AND USER'S DATA.
--------------------------------------------------------------
NCLUDE 9,CMNGUSSI.FTN/G
   LOGICAL LOGIC1,LOGIC
   DIMENSION LOGIC(100)
   COMMON/LDATA/LOGIC1(309)
   EQUIVALENCE (LOGIC(1),LOGIC1(210))
   DIMENSION INTGR(100)
   COMMON/IDATA/INTGR1(194)
   EQUIVALENCE (INTGR(1),INTGR1(95))
   DIMENSION RE(100)
   COMMON/RDATA/RE1(421)
   EQUIVALENCE (RE(1),RE1(322))
   COMMON/BOUND/LOCREG(60),
  &TR1,CP1R1(7),VP1R1(7),CP2R1(5),VP2R1(5),CPNR1(5),VPNR1(5),
  &TR2,CP1R2(7),VP1R2(7),CP2R2(5),VP2R2(5),CPNR2(5),VPNR2(5),
  &TR3,CP1R3(7),VP1R3(7),CP2R3(5),VP2R3(5),CPNR3(5),VPNR3(5),
  &TR4,CP1R4(7),VP1R4(7),CP2R4(5),VP2R4(5),CPNR4(5),VPNR4(5),
  &TR5,CP1R5(7),VP1R5(7),CP2R5(5),VP2R5(5),CPNR5(5),VPNR5(5),
  &TR6,CP1R6(7),VP1R6(7),CP2R6(5),VP2R6(5),CPNR6(5),VPNR6(5),
  &TR7,CP1R7(7),VP1R7(7),CP2R7(5),VP2R7(5),CPNR7(5),VPNR7(5),
  &TR8,CP1R8(7),VP1R8(7),CP2R8(5),VP2R8(5),CPNR8(5),VPNR8(5),
  &TR9,CP1R9(7),VP1R9(7),CP2R9(5),VP2R9(5),CPNR9(5),VPNR9(5),
  &TR10,CP1R10(7),VP1R10(7),CP2R10(5),VP2R10(5),CPNR10(5),VPNR10(5)
   DIMENSION TCVREG(350)
   EQUIVALENCE (TCVREG(1),TR1)
NCLUDE 9,GUSSEQUI.FTN/G
     LOGICAL CARTES,POLAR,SPDATA,SKEW,TWODYZ,ONEDZ,STOVAR(25),
  &    SOLVAR(25),PRINT(25),RESID(25),CMPRSS,CONEMU,LSP1,
  &    CONRHO,EMDOT,ONEPHS,INCORE(10),ISAVED,SAVEI,SAVEM,
  &    RESTRT,XCYCLE,MONITR,REGION(10),STEADY,WHOLEP,SLABPP,
  &    RAIN,BLOCKZ,PWSTAG,RUN(30),PLOT,RESMAP,FLAG,BLOCK,
  &    TEST,CONC1(4),DISTIL
   LOGICAL SPRESS,PARAB,DONACC,OVERLY,SACC,
  &    GUSSIE,CATHY,CONNIE,CORA,ESTER,FLASH,FLORA,
  &    FOCS,GENMIX,HESTER,PICALO,PLANT,SPLASH,HELGA,
  &    TACT,TIBALT,TOPSI,PAMELA,TABLES,WSTAG,
  &    CONMOD,GROSTA,SUBPST,SUBWGR
   EQUIVALENCE (CARTES,LOGIC1(1)),(POLAR,LOGIC1(2)),
  &    (SPDATA,LOGIC1(3)),(SKEW,LOGIC1(4)),(TWODYZ,LOGIC1(5)),
  &    (ONEDZ,LOGIC1(6)),(STOVAR(1),LOGIC1(7)),
  &    (SOLVAR(1),LOGIC1(32)),(PRINT(1),LOGIC1(57)),
  &    (RESID(1),LOGIC1(82)),(CMPRSS,LOGIC1(107)),
  &    (CONEMU,LOGIC1(108)),(LSP1,LOGIC1(109)),
  &    (CONRHO,LOGIC1(110)),(EMDOT,LOGIC1(111)),
  &    (ONEPHS,LOGIC1(112)),(INCORE(1),LOGIC1(113))
   EQUIVALENCE(ISAVED,LOGIC1(123)),(SAVEI,LOGIC1(124)),
  &    (SAVEM,LOGIC1(125)),(RESTRT,LOGIC1(126)),
  &    (XCYCLE,LOGIC1(127)),(MONITR,LOGIC1(128)),
  &    (REGION(1),LOGIC1(129)),(STEADY,LOGIC1(139)),
  &    (WHOLEP,LOGIC1(140)),(SLABPP,LOGIC1(141)),
  &    (RAIN,LOGIC1(142)),(BLOCKZ,LOGIC1(143)),
  &    (PWSTAG,LOGIC1(144)),(RUN(1),LOGIC1(145)),
  &    (PLOT,LOGIC1(175)),(RESMAP,LOGIC1(176)),
  &    (FLAG,LOGIC1(177)),(BLOCK,LOGIC1(178)),(TEST,LOGIC1(179))
   EQUIVALENCE(CONC1(1),LOGIC1(206))
   EQUIVALENCE (SPRESS,LOGIC(100)),(PARAB,LOGIC(99)),
  &    (DONACC,LOGIC(98))
   EQUIVALENCE (OVERLY,LOGIC(96)),(SACC,LOGIC(95)),
  &    (WSTAG,LOGIC(94))
   EQUIVALENCE (DISTIL,LOGIC(90))
   EQUIVALENCE (PAMELA,LOGIC(67)),(TOPSI,LOGIC(66)),
  &    (TIBALT,LOGIC(65)),(TACT,LOGIC(64)),(HELGA,LOGIC(63)),
  &    (SPLASH,LOGIC(62)),(PLANT,LOGIC(61)),(PICALO,LOGIC(60)),
```

```
     &   (HESTER,LOGIC(59)),(GENMIX,LOGIC(58)),(FOCS,LOGIC(57)),        MEL00730
     &   (FLORA,LOGIC(56)),(FLASH,LOGIC(55)),(ESTER,LOGIC(54)),         MEL00740
     &   (CORA,LOGIC(53)),(CONNIE,LOGIC(52)),(CATHY,LOGIC(51)),         MEL00750
     &   (GUSSIE,LOGIC(50))                                             MEL00760
      EQUIVALENCE (TABLES,LOGIC(48)),(CONMOD,LOGIC(47)),                MEL00770
     &   (GROSTA,LOGIC(46)),(SUBPST,LOGIC(45))                          MEL00780
      EQUIVALENCE (SUBWGR,LOGIC(43))                                    MEL00790
      INTEGER FSTEP,FSWEEP,TSTSWP,ITAB(8),MTABVR(8)                     MEL00800
      DIMENSION ISPCSO(25),LITER(25)                                    MEL00810
      EQUIVALENCE (NX,INTGR1(1)),(NY,INTGR1(2)),                        MEL00820
     &   (NZ,INTGR1(3)),(ISPCSO(1),INTGR1(4)),                          MEL00830
     &   (NREGN,INTGR1(29)),(NPHI,INTGR1(30)),                          MEL00840
     &   (LITKE,INTGR1(31)),(LITHYD,INTGR1(32)),                        MEL00850
     &   (LITH,INTGR1(33)),(LITCNC,INTGR1(34)),                         MEL00860
     &   (LITSLB,INTGR1(35)),(NRUN,INTGR1(36)),                         MEL00870
     &   (LITER(1),INTGR1(37)),(FSTEP,INTGR1(62)),                      MEL00880
     &   (FSWEEP,INTGR1(63)),(LSTEP,INTGR1(64))                         MEL00890
      EQUIVALENCE (LSWEEP,INTGR1(65)),(NPRINT,INTGR1(66)),              MEL00900
     &   (IERRP,INTGR1(67)),(IMAXP,INTGR1(68)),                         MEL00910
     &   (IEMU1,INTGR1(69)),(IXMON,INTGR1(70)),                         MEL00920
     &   (IYMON,INTGR1(71)),(IZMON,INTGR1(72)),                         MEL00930
     &   (KEMU,INTGR1(73)),(KMAIN,INTGR1(74)),                          MEL00940
     &   (KINDEX,INTGR1(75)),(KGEOM,INTGR1(76)),                        MEL00950
     &   (KINPUT,INTGR1(77)),(KSODAT,INTGR1(78))                        MEL00960
      EQUIVALENCE(KFLAG,INTGR1(79)),(KCOMPF,INTGR1(80)),                MEL00970
     &   (KSORCE,INTGR1(81)),(KLES1D,INTGR1(82)),                       MEL00980
     &   (KLES2D,INTGR1(83)),(KLES3D,INTGR1(84)),                       MEL00990
     &   (KCOMPP,INTGR1(85)),(KADJST,INTGR1(86)),                       MEL01000
     &   (KFLUX,INTGR1(87)),(KSHIFT,INTGR1(88)),                        MEL01010
     &   (KOUTPT,INTGR1(89)),(KDIF,INTGR1(90)),                         MEL01020
     &   (KCOMPU,INTGR1(91)),(KCOMPV,INTGR1(92)),                       MEL01030
     &   (KCOMPW,INTGR1(93)),(KCOMPR,INTGR1(94))                        MEL01040
      EQUIVALENCE (KELIN,INTGR(100)),(MEANDF,INTGR(99)),                MEL01050
     &   (NUMCLS,INTGR(98)),                                            MEL01060
     &   (IRHO1,INTGR(97)),(IRHO2,INTGR(96)),                           MEL01070
     &   (IZW1,INTGR(95)),(IZW2,INTGR(94)),                             MEL01080
     &   (MGRID,INTGR(93)),(KWALL,INTGR(92)),                           MEL01090
     &   (IZPR1,INTGR(91)),(IZPR2,INTGR(90)),                           MEL01100
     &   (ISTPR1,INTGR(89)),(ISTPR2,INTGR(88)),                         MEL01110
     &   (NTPRIN,INTGR(87))                                             MEL01120
      EQUIVALENCE (IMDOT,INTGR(86)),(IHSAT,INTGR(85)),                  MEL01130
     &   (ICFIP,INTGR(84)),(NTABLE,INTGR(83)),(NTABVR,INTGR(82)),       MEL01140
     &   (LINTAB,INTGR(81)),(NPRTAB,INTGR(80)),(NMON,INTGR(79)),        MEL01150
     &   (ITAB(1),INTGR(71)),(MTABVR(1),INTGR(63)),                     MEL01160
     &   (NZPRIN,INTGR(62)),                                            MEL01170
     &   (NXPRIN,INTGR(61)),(NYPRIN,INTGR(60)),                         MEL01180
     &   (LDISTL,INTGR(59)),(TSTSWP,INTGR(58)),                         MEL01190
     &   (KDBEXP,INTGR(57)),(KDBRHO,INTGR(56)),                         MEL01200
     &   (KDBMDT,INTGR(55))                                             MEL01210
      EQUIVALENCE (ILOOP1,INTGR(53)),(ILOOPN,INTGR(52)),                MEL01220
     &   (IPBP,INTGR(51)),(IZPRF,INTGR(50)),                            MEL01230
     &   (IZPRL,INTGR(49)),(ISTPRF,INTGR(48)),                          MEL01240
     &   (ISTPRL,INTGR(47)),(KDBGEN,INTGR(46)),                         MEL01250
     &   (IVELF,INTGR(45)),(IVELL,INTGR(44)),                           MEL01260
     &   (IKEF,INTGR(43)),(IKEL,INTGR(42)),                             MEL01270
     &   (IENTF,INTGR(41)),(IENTL,INTGR(40)),                           MEL01280
     &   (ICNCF,INTGR(39)),(ICNCL,INTGR(38)),                           MEL01290
     &   (NVEL,INTGR(37)),(NKE,INTGR(36)),                              MEL01300
     &   (NENT,INTGR(35)),(NCNC,INTGR(34)),                             MEL01310
     &   (NZSTP,INTGR(33)),(NPRMNT,INTGR(32))                           MEL01320
      DIMENSION SPARE1(20),XUDIST(30),YVDIST(30),ZWDIST(50),            MEL01330
     &   SIGMA(25),CRIT(25),DTFALS(25),RESREF(25),                      MEL01340
     &   TITLE(25),FIINIT(25),TFRAC(30)                                 MEL01350
      EQUIVALENCE (SPARE1(1),RE1(1)),                                   MEL01360
     &   (TFRAC(1),RE1(21)),(XULAST,RE1(51)),                           MEL01370
     &   (YVLAST,RE1(52)),(ZWLAST,RE1(53)),                             MEL01380
     &   (XUDIST(1),RE1(54)),(YVDIST(1),RE1(84)),                       MEL01390
     &   (ZWDIST(1),RE1(114)),(SIGMA(1),RE1(164)),                      MEL01400
     &   (CRIT(1),RE1(189)),(DTFALS(1),RE1(214)),                       MEL01410
     &   (RESREF(1),RE1(239)),(EMU1,RE1(264)),                          MEL01420
     &   (RHO1,RE1(265)),(RHO2,RE1(266)),                               MEL01430
     &   (TLAST,RE1(267)),(CFIPS,RE1(268))                              MEL01440
```

```
  EQUIVALENCE (AMDOT,RE1(269)),(FIINIT(1),RE1(270)),
 &    (RELAXP,RE1(295)),(TITLE(1),RE1(296)),
 &    (DT,RE1(321)),
 &    (RINNER,ZWDIST(50)),(SNALFA,ZWDIST(49)),
 &    (PBAR,ZWDIST(48))
  EQUIVALENCE (SLOEMU,RE(100)),(SLORHO,RE(99)),(RLXRHO,RE(98)),
 &    (RHOMAX,RE(97)),(RHOMIN,RE(96)),
 &    (EMUMAX,RE(95)),(EMUMIN,RE(94)),
 &    (TKEMAX,RE(93)),(TKEMIN,RE(92)),
 &    (RLXPZ,RE(91)),
 &    (ABUOY,RE(90)),(HREF,RE(89)),
 &    (AGRAVX,RE(88)),(AGRAVY,RE(87)),
 &    (AGRAVZ,RE(86)),(ATIME,RE(85))
  EQUIVALENCE (RLXPXY,RE(84)),
 &    (EPSMAX,RE(83)),(EPSMIN,RE(82)),
 &    (ARHO1,RE(81)),(BRHO1,RE(80)),(CRHO1,RE(79))
  EQUIVALENCE (AZW2,RE(73)),(BZW2,RE(74)),
 &    (PINT,RE(75)),(CZW2,RE(76)),
 &    (DZ,RE(78)),(ZWL,RE(77)),(ZW2M1T,RE(72)),
 &    (VELMIN,RE(70)),(VELMAX,RE(69)),
 &    (FALCOM,RE(68)),(CMDOT,RE(67)),
 &    (RLXMDT,RE(66)),(AMDTMX,RE(65)),(AMDTMN,RE(64)),
 &    (RADMAX,RE(63)),(RADMIN,RE(62)),
 &    (PRESS,RE(61)),(CP1,RE(60)),
 &    (CP2,RE(59)),(CP3,RE(58)),(FSTOIC,RE(57)),
 &    (ARRCON,RE(56)),(PREEXP,RE(55)),(CA1I,RE(54))
  EQUIVALENCE (CA2I,RE(53)),(CA13,RE(52)),(CA23,RE(51)),
 &    (H1SAT,RE(50)),(H2SAT,RE(49)),
 &    (EMULAM,RE(48))
  EQUIVALENCE (TWPRCN,RE(42)),(PSATEX,RE(41))
  EQUIVALENCE(ELEXP,RE(39)),(EWALL,RE(38)),
 &    (SIGE,RE(37)),(SIGK,RE(36)),(TAUDK,RE(35))
  DIMENSION LOC1(6),LOC2(6),LOC3(6),LOC4(6),LOC5(6),
 &    LOC6(6),LOC7(6),LOC8(6),LOC9(6),LOC10(6)
  EQUIVALENCE (LOCREG(1),LOC1(1)),(LOCREG(7),LOC2(1))
  EQUIVALENCE (LOCREG(13),LOC3(1)),(LOCREG(19),LOC4(1))
  EQUIVALENCE (LOCREG(25),LOC5(1)),(LOCREG(31),LOC6(1))
  EQUIVALENCE (LOCREG(37),LOC7(1)),(LOCREG(43),LOC8(1))
  EQUIVALENCE (LOCREG(49),LOC9(1)),(LOCREG(55),LOC10(1))
  DIMENSION XUFRAC(30),YVFRAC(30),ZWFRAC(30),ENDIT(25)
  DIMENSION XFRAC(30),YFRAC(30),ZFRAC(30)
  EQUIVALENCE(XUDIST(1),XUFRAC(1),XFRAC(1)),
 &           (YVDIST(1),YVFRAC(1),YFRAC(1)),
 &            (ZWDIST(1),ZWFRAC(1),ZFRAC(1)),(LITCNC,LITC),
 &            (ENDIT(1),CRIT(1)),(RLXP,RELAXP)
  EQUIVALENCE (NPRMON,LITER(1))
CLUDE 9,CMNGRFIC.FTN/G
  LOGICAL  GRAPHS, ORTHOG, ANTSYM
  COMMON / GRAF3 / GRAPHS, ORTHOG, ANTSYM, NPRT
  REAL NAME
  COMMON / GRAF4 / NAME (25),ITITL (5)
  COMMON/CPI/IPWRIT,IDUM(243)
  DIMENSION GDTAPE(3),DFAULT(4)
  DIMENSION ARRAY1(309),ARRAY2(194),ARRAY3(421)
  LOGICAL ARRAY1,LSPDA,WRT,RD,NAMLST
  INTEGER ARRAY2,XPLANE,YPLANE,ZPLANE
  INTEGER P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,EP,H1,H2,H3,C1,C2,
 &C3,C4
  REAL NORTH,LOW
  EQUIVALENCE (ARRAY1(1),CARTES),(ARRAY2(1),NX)
  EQUIVALENCE (ARRAY3(1),SPARE1(1)),(M1,R1),(M2,R2)
  EQUIVALENCE (LSTRUN,INTGR(12)),(NAMLST,LOGIC(88))
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS:
  GRAFFIC ARRAYS DIMENSIONED AS NEEDED...
  COMMON/GRAF1/PHI1(1) /GRAF2/PHI2(1)
  POROSITY & SPECIAL DATA ARRAYS DIMENSIONED AS NEEDED...
  DIMENSION PE(1,1,1),PN(1,1,1),PH(1,1,1),PC(1,1,1)
  DIMENSION LSPDA(1),ISPDA(1),RSPDA(1)
  USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.
  DATA TITEMO/4HTEMO/
  DATA TITEMP/4HTEMP/
```

```
      DATA TITDEN/4HRHO1/
      DATA NLSP,NISP,NRSP/1,1,1/
C     USER PLACES HIS DATA STATEMENTS HERE.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS:
C-----------------------------------------------------------------
CHAPTER 2   SET CONSTANTS, AND ARRANGE FILE MANIPULATIONS.
C----------------------------------------------------------------
C     PLEASE DO NOT ALTER, OR RE-SET, ANY OF THE REMAINING
C     STATEMENTS OF THIS CHAPTER.
      DATA CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME/
     & 0.,1.,2.,3.,4.,5.,6.,7. /
      DATA P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,KE,EP,H1,H2,H3,C1,C2,
     &C3,C4/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20/
      DATA FIXFLU,FIXVAL,ONLYMS,WALL/1.E-10,1.E10,0.0,-10.0/
      DATA IPLANE,XPLANE,YPLANE,ZPLANE/0,1,2,3/
      DATA WRT,RD,DFAULT/.TRUE.,.FALSE.,4HDEFA,4HULT.,4HDTA/,1HG/
      DATA GDTAPE/4HGUSI,4HE1.D,2HTA/
      DATA NLDATA,NIDATA,NRDATA/309,194,421/
      DATA NLCREG,NTCVRG/60,350/
CD    CALL SETGRF
      CALL TAPES(10,GDTAPE,3,1,4*NRDATA)
C------------------------------READ DEFAULT FILE IF BLOCKDATA ABSENT
          IF(INTGR1(29).NE.10) GO TO 2
      CALL WRIT40(40HDATA ESTABLISHED IN BLOCK DATA.          )
      GO TO 3
    2 CALL DEFLT
CD  2 CALL TAPES(1,DFAULT,4,2,4*NRDATA)
CD    CALL DATAIO(RD,1)
      CALL WRIT40(40HDATA TAKEN FROM DEFAULT.DTA ON GROUP A/C)
    3 CALL WRIT40(40HFILE MODSTL.FTN IS THE SATLIT USED.      )
C-----------------------------------------------------------------
CHAPTER 3   DEFINE DATA FOR NRUN RUNS.
C----------------------------------------------------------------
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS:
C---  GROUP  41 MULTI-RUNS : RUN(1-30)<.T.,29*.F.>
C
      DO 410 II=1,1
  410 RUN(II)=.TRUE.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 STARTS:
        DO 10 IRUN=1,30
          IF(.NOT.RUN(IRUN)) GO TO 10
      NRUN=NRUN+1
      LSTRUN=IRUN
   10 CONTINUE
        DO 999 IRUN=1,LSTRUN
          IF(.NOT.RUN(IRUN)) GO TO 999
      INTGR(11) = IRUN
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 3 STARTS:
C---  ALL INTEGER VARIABLES ARE DEFAULTED TO 0, AND REAL VARIABLES
C     TO 0.0, UNLESS OTHERWISE INDICATED.
C     E.G. BY VARIABLE<10>, OR <10.0> AS APPROPRIATE.
C     THE DEFAULT SETTINGS OF ALL LOGICAL VARIABLES ARE ALWAYS
C     INDICATED, E.G. VARIABLE<.T.>, OR  VARIABLE<.F.>.
C
C---  RUN1
C------------------   ---------------   -------------------
C---  GROUP 1. FLOW TYPE :
C     PARAB<.F.>,CARTES<.T.>,ONEPHS<.T.>
      ONEPHS=.FALSE.
C------------------   ---------------   -------------------
C---  GROUP 2. TRANSIENCE :
C     STEADY<.T.>,ATIME,LSTEP<1>,FSTEP<1>
C     TLAST<1.E10>,TFRAC(1-30)<30*1.>
      STEADY=.FALSE.
C     LSTEP=30
C     ATIME=0.
C     TLAST=50E-4
C     SERVICE SUBROUTINE FOR 'NT' POWER-LAW TIME STEPS:
```

```
   CALL GRDPWR(0,30,50E-4,1.)
----------------     --------------     ------------------
-  GROUP 3. X-DIRECTION :
   NX<1>,XULAST<1.0>,XFRAC(1-30)
   SERVICE SUBROUTINE FOR POWER-LAW GRID:
   CALL GRDPWR(1,NX,XULAST,1.)
----------------     --------------     ------------------
-  GROUP 4. Y-DIRECTION :
   NY<1>,YVLAST<1.0>,YFRAC(1-30),RINNER,SNALFA
   SERVICE SUBROUTINE FOR POWER-LAW GRID:
   NY=100
   CALL GRDPWR(2,100,4.E-3,1.   )
----------------     --------------     ------------------
-  GROUP 5. Z-DIRECTION :
   NZ<1>,ZWLAST<1.0>,ZFRAC(1-30)
   SERVICE SUBROUTINE FOR POWER-LAW GRID:
   CALL GRDPWR(3,NZ,ZWLAST,POWER)
----------------     --------------     ------------------
-  GROUP 6. MOVING GRID :
   MGRID,IZW1,IZW2,AZW2,BZW2,CZW2,PINT,ZW2MIT
----------------     --------------     ------------------
-  GROUP 7. BLOCKAGE: BLOCK<.F.>,IPLANE,IPWRIT
  *SET CONSTANT POROSITIES OVER SUB-DOMAINS USING:
   CALL CONPOR(IR,TYPE,VALUE,IXF,IXL,IYF,IYL,IZF,IZL), WHERE:
   IR=RUN SECTION NUMBER, E.G. 1 FOR RUN1 SECTION; 'TYPE'= EAST,
   WEST, NORTH, SOUTH, HIGH, LOW & CELL. 'VALUE'=WANTED POROSITY
   OVER REGION IXF,...IZL.
  *DIMENSION ARRAYS PE(NX,NY,NZ), PN(NX,NY,NZ), PH(NX,NY,NZ), &
   PC(NX,NY,NZ) ABOVE.
  *FOR FULLY-BLOCKED CELLS (IE. 'VALUE'= 0.0) USER NEED SET ONLY
   THE 'CELL' POROSITY (TO ZERO), AS CELL-FACE AREAS ARE THEN
   AUTOMATICALLY ZEROED.
  *FOR SATELLITE PRINTOUT OF ALL POROSITIES IN DOMAIN, 'IPLANE'=
   XPLANE YPLANE OR ZPLANE, FOR DESIRED CROSS-SECTION DIRECTION.
  *FOR EACH 'TYPE' A MAXIMUM OF 10 CALLS TO CONPOR IS ALLOWED,
   BUT IF REQUIREMENTS EXCEED THIS PROVISION SET BLOCK=.T. &
   IPWRIT=-1, AND SET POROSITY ARRAYS EXPLICITLY HERE AS WANTED.
   IN THIS CASE, THE USER  M U S T   SET  A L L  ELEMENTS OF
   ARRAYS PE, PN, PH, PC (MANY MAY BE 0.0 OR 1.0). HE MAY USE:
   CALL CR(PARRAY,VALUE,IXF,IXL,IYF,IYL,IZF,IZL,NX,NY,NZ)
   ANY NUMBER OF TIMES, TO SET 'PARRAY' (= PE, ETC.) TO
   'VALUE' OVER RANGE IXF TO IXL, IYF TO IYL, IZF TO IZL.
  *CONPOR  M U S T  N O T  BE USED IN CONJUNCTION WITH EXPLICIT
   SETTINGS OF THE ARRAYS (INCLUDING SETTINGS VIA CR).
----------------     --------------     ------------------
-  GROUP 8.DEPENDENT VARIABLES TO BE SOLVED FOR OR STORED :
   SOLVAR(1-25)<25*.F.>,STOVAR(1-25)<25*.F.>,CONC1(1-4)<4*.T.>
   USE FOLLOWING NAMED INTEGERS FOR ARRAY ELEMENTS 1-20:
   P1,PP,U1,U2,V1,V2,W1,W2,M1,M2,RS,KE,EP,H1,H2,H3,C1,C2,C3,C4.
   SOLVAR (P1)=.TRUE.
   SOLVAR (PP)=.TRUE.
   SOLVAR (V1)=.TRUE.
   SOLVAR (V2)=.TRUE.
   SOLVAR (R1)=.TRUE.
   SOLVAR (R2)=.TRUE.
   SOLVAR (RS)=.TRUE.
   SOLVAR (H1)=.TRUE.
   SOLVAR (H2)=.TRUE.
   SOLVAR (C1)=.TRUE.
   SOLVAR (C2)=.TRUE.
   SOLVAR (C3)=.TRUE.
   STOVAR (U1)=.TRUE.
   STOVAR (U2)=.TRUE.
   STOVAR (24)=.TRUE.
----------------     --------------     ------------------
-  GROUP 9. VARIABLE LABELS :
   TITLE(1-25)<2HP1,2HPP,2HU1,2HU2,2HV1,2HV2,2HW1,2HW2,2HR1,
   2HR2,2HRS,2HKE,2HEP,2HH1,2HH2,2HH3,2HC1,2HC2,
   2HC3,2HC4,2HRX,2HRY,2HRZ, 2*4H****>
   TITLE(U1)=TITEMO
   TITLE(U2)=TITEMP
   TITLE(PP)=TITDEN
```

```
C------------------   ---------------   -------------------
C---  GROUP 10 PROPERTIES:
C     IRHO1<1>,IRHO2<1>,RHO1<1.0>,RHO2<1.0>,
C     ARHO1<1.0>,BRHO1<1.0>,CRHO1<1.0>
C     IEMU1<1>,EMU1<1.0>,EMULAM<1.E-10>
C     IHSAT,H1SAT,H2SAT,PSATEX<1.0>
C     SIGMA(1-25)<1.0,2.0,1.,1.E10,1.,1.E10,1.,1.E10,
C     4*1.0,1.314,1.0,1.E10,10*1.0>
      IRHO1=-1
      IEMU1=-1
      IRHO2=-1
      SIGMA(6)=1.E10
      SIGMA(15)=1.E10
      SIGMA(H1)=-1
      SIGMA(C1)=-1
      SIGMA(C2)=-1
      SIGMA(C3)=-1
C------------------   ---------------   -------------------
C---  GROUP 11 INTER-PHASE TRANSFER PROCESSES :
C     ICFIP,CFIPS,IMDOT,CMDOT,CA1I<1.E6>,CA2I<1.E6>
      ICFIP= 1
      CFIPS=5E07
      IMDOT=-1
C------------------   ---------------   -------------------
C---  GROUP 12 SPECIAL SOURCES :
C     ISPCSO(1-25),AGRAVX,AGRAVY,AGRAVZ,ABUOY,HREF
C------------------   ---------------   -------------------
C---  GROUP 13 INITIAL FIELDS :
C     FIINIT(1-25)<25*1.E-10>
      FIINIT(6)=1.E-10
      FIINIT(15)=1.E-10
      FIINIT(1) =100000.
      FIINIT(5)=10101
      FIINIT(9)=10101
      FIINIT(14)=10101
      FIINIT(17)=10101
      FIINIT(18)=10101
      FIINIT(19)=10101
      FIINIT(10)=0.0
C------------------   ---------------   -------------------
C---  GROUP 14 BOUNDARY/INTERNAL CONDITIONS :
C     ILOOP1,ILOOPN,XCYCLE<.F.>,PBAR,REGION(1-10)<10*.T.>
C    *N.B. ALL 10 REGIONS ARE DEFAULTED .TRUE..  THE USER SHOULD
C     SET REGION(I)=.FALSE. FOR UNUSED REGIONS 'I'.
      DO 140 I=1,10
  140 REGION(I)=.FALSE.
C------------------   ---------------   -------------------
C---  GROUP 15 TO 24; REGIONS 1 TO 10
C---  ONLY THOSE REGIONS ARE ACTIVE WHICH ARE SPECIFIED BY THE
C     USER, PREFERABLY BY WAY OF:-
C     CALL PLACE(IREGN,TYPE,IXF,IXL,IYF,IYL,IZF,IZL) &
C     CALL COVAL(IREGN,VARBLE,COEFF,VALUE)
      CALL PLACE(1,SOUTH,1,NX,1,1,1,NZ)
      CALL PLACE(3,NORTH,1,NX,100,100,1,NZ)
      CALL COVAL(3,M1,FIXVAL,100000.)
      CALL COVAL(3,M2,FIXVAL,100000.)
C------------------   ---------------   -------------------
C---  GROUP 25 GROUND STATION :
C     GROSTA<.F.>,NAMLST<.F.>
      GROSTA=.TRUE.
C    *NAMLST ACTIVATES NAMELIST IN GROUND.
C------------------   ---------------   -------------------
C---  GROUP 26 SOLUTION TYPE AND RELATED PARAMETERS :
C     WHOLEP<.F.>,SUBPST<.F.>,DONACC<.F.>
C------------------   ---------------   -------------------
C---  GROUP 27 SWEEP AND ITERATION NUMBERS :
C     FSWEEP<1>,LSWEEP<1>,LITHYD<1>,LITC<1>,LITKE<1>,LITH<1>,
      LSWEEP=20
C     LITER(1-25)<9*1,-1,15*1>
      LITER(2)= 50
      LITER(17)=20
      LITER(18)=20
```

```
    LITER(19)=20
    IVELF<1>,NVEL<1>,IVELL<10000>,
    IKEF<1>,NKE<1>,IKEL<10000>,
    IENTF<1>,NENT<1>,IENTL<10000>,
    ICNCF<1>,NCNC<1>,ICNCL<10000>,
    IRHO1F<1>,NRHO1<1>,IRHO1L<10000>,
    IRHO2F<1>,NRHO2<1>,IRHO2L<10000>
-----------------   --------------   ------------------
--  GROUP 28 TERMINATION CRITERIA :
    ENDIT(1-25)<9*1.E-10,0.5,15*1.E-10>
-----------------   --------------   ------------------
--  GROUP 29 RELAXATION  :
    RLXP<1.>,RLXPXY<1.>,RLXPZ<1.>,RLXRHO<1.>,RLXMDT<1.>,
    RLXRHO=0.2
    DTFALS(C1)=4.E-5
    DTFALS(C2)=4.E-5
    DTFALS(C2)=4.E-5
    DTFALS(3-25)<23*1.E10>
-----------------   --------------   ------------------
--  GROUP 30 LIMITS :
    VELMAX<1.E10>,VELMIN<-1.E10>,RHOMAX<1.E10>,RHOMIN<1.E-10>,
    TKEMAX<1.E10>,TKEMIN<1.E-10>,EMUMAX<1.E10>,EMUMIN<1.E-10>,
    EPSMAX<1.E10>,EPSMIN<1.E-10>,AMDTMX<1.E10>,AMDTMN<-1.E10>
-----------------   --------------   ------------------
-   GROUP 31 SLOWING DEVICES : SLORHO<1.>,SLOEMU<1.>
-----------------   --------------   ------------------
-   GROUP 32 PRINT-OUT OF VARIABLES :
    PRINT(1-25)<.T.,.F.,23*.T.>,SUBWGR<.F.>
    PRINT(U1)=.TRUE.
    PRINT(U2)=.TRUE.
    PRINT(PP)=.TRUE.
    PRINT(24)=.TRUE.
-----------------   --------------   ------------------
-   GROUP 33 MONITOR PRINT-OUT :
    IYMON=25
    IXMON<1>,IYMON<1>,IZMON<1>,NPRMON<1>,NPRMNT<1>
-----------------   --------------   ------------------
-   GROUP 34 FIELD PRINT-OUT CONTROL :
    NPRINT<100>,NTPRIN<100>,NXPRIN<1>,NYPRIN<1>,NZPRIN<1>,
    IZPRF<1>,ISTPRF<1>,IZPRL<10000>,ISTPRL<10000>
    NUMCLS<10>,KOUTPT
    KOUTPT=-1
    NTPRIN=1
-----------------   --------------   ------------------
-   GROUP 35 TABLE CONTROL :
    TABLES<.F.>,NTABLE,NTABVR,LINTAB,NPRTAB,NMON,
    ITAB(1-8),MTABVR(1-8)
-----------------   --------------   ------------------
    GROUP 36-38 ARE NOT DOCUMENTED IN THE INSTRUCTION
    MANUAL AND ARE INTENDED FOR MAINTENANCE PURPOSES ONLY
-   GROUP 36 DEBUG PRINT-OUT SLAB AND TIME-STEP :
    IZPR1<1>,IZPR2<1>,ISTPR1<1>,ISTPR2<1>
-----------------   --------------   ------------------
-   GROUP 37 DEBUG SWEEP AND SUBROUTINES :
    KEMU,KMAIN,KINDEX,KGEOM,KINPUT,KSODAT,KCOMPF,KSORCE,
    KSOLV1,KSOLV2,KSOLV3,KCOMPP,KADJST,KFLUX,KSHIFT,KDIF,
    KCOMPU,KCOMPV,KCOMPW,KCOMPR,KWALL,KDBRHO<-1>,KDBEXP,KDBMDT
    KDBGEN
-----------------   --------------   ------------------
-   GROUP 38 MONITOR,TEST,AND FLAG :
    MONITR<.F.>,FLAG<.F.>,TEST<.T.>,KFLAG<1>
    END OF MAINTENANCE-ONLY SECTION
-----------------   --------------   ------------------
-   GROUP 39 ERROR AND RESIDUAL PRINT-OUT :
    IERRP<1000>,RESREF(1,3-24)<25*1.>,RESMAP<.F.>,
    RESID(1-25)<2*.F.,23*.T.>,KOUTPT
    IERRP=LSWEEP
    RESREF(1)=5.E-5
    RESREF(V1)=5.E-5
    RESREF(V2)=5.E-5
    RESREF(H1)=40.
    RESREF(H2)=40.
```

```
      RESREF(C1)=5.E-5
      RESREF(C2)=5.E-5
      RESREF(C3)=5.E-5
C-------------------    ---------------    -------------------
C---  GROUP 40 SPECIAL DATA : LOGIC(1..10),INTGR(1..10),RE(21..30),
C     NLSP<1>,NISP<1>,NRSP<1>,SPDATA<.F.>,LSPDA(1),ISPDA(1),RSPDA(1)
C     USE FIRST 10 ELEMENTS OF ARRAYS LOGIC & INTGR AND 21ST
C     TO 30TH OF ARRAY RE FOR TRANSFERRING SPECIAL DATA FROM
C     SATELLITE TO GROUND, BUT IF REQUIREMENTS EXCEED THIS
C     PROVISION SET SPDATA = .T., AND DIMENSION ARRAYS LSPDA,
C     ISPDA, RSPDA ABOVE AND IN GROUND AS NEEDED, AND SET HERE
C     NLSP, NISP, NRSP TO DIMENSION VALUES.
C-------------------    ---------------    -------------------
C---  GROUP 42 RESTARTS AND DUMPS : SAVEM<.F.>,RESTRT<.F.>,KINPUT
C-------------------    ---------------    -------------------
C---  GROUP 43 GRAFFIC :
C     GRAPHS<.F.>,ORTHOG<.T.>,ANTSYM,NPRT<1>,ITITL<5*4H****>
C---  FOR A GRAFFIC RUN, DIMENSION PHI1 & PHI2 AS FOLLOWS:
C     PHI1(NX*NY*NZ*NM)
C     PHI2((NX+2)*(NY+2)*(NZ+2)*(NM+IBLK)) , WHERE
C     NM=NO. OF VARIABLES STORED + DENSITY(-IES)
C     IBLK=0 IF BLOCK=.FALSE.,=4 IF A 3D RUN,
C     =3 IF A 2D.YZ RUN.
C-------------------    ---------------    -------------------
          IF(IRUN.EQ.1) GO TO 900
C---  RUN2
          IF(IRUN.EQ.2) GO TO 900
C---  RUN3
          IF(IRUN.EQ.3) GO TO 900
C---  RUN4
          IF(IRUN.EQ.4) GO TO 900
C---  RUN5
          IF(IRUN.EQ.5) GO TO 900
C---  RUN6
          IF(IRUN.EQ.6) GO TO 900
C---  RUN7
          IF(IRUN.EQ.7) GO TO 900
C---  RUN8
          IF(IRUN.EQ.8) GO TO 900
C---  RUN9
          IF(IRUN.EQ.9) GO TO 900
C---  RUN10
          IF(IRUN.EQ.10) GO TO 900
C---  RUN11
          IF(IRUN.EQ.11) GO TO 900
C---  RUN12
          IF(IRUN.EQ.12) GO TO 900
C---  RUN13
          IF(IRUN.EQ.13) GO TO 900
C---  RUN14
          IF(IRUN.EQ.14) GO TO 900
C---  RUN15
          IF(IRUN.EQ.15) GO TO 900
C---  RUN16
          IF(IRUN.EQ.16) GO TO 900
C---  RUN17
          IF(IRUN.EQ.17) GO TO 900
C---  RUN18
          IF(IRUN.EQ.18) GO TO 900
C---  RUN19
          IF(IRUN.EQ.19) GO TO 900
C---  RUN20
          IF(IRUN.EQ.20) GO TO 900
C---  RUN21
          IF(IRUN.EQ.21) GO TO 900
C---  RUN22
          IF(IRUN.EQ.22) GO TO 900
C---  RUN23
          IF(IRUN.EQ.23) GO TO 900
C---  RUN24
          IF(IRUN.EQ.24) GO TO 900
C---  RUN25
```

```
      IF(IRUN.EQ.25) GO TO 900
-  RUN26
      IF(IRUN.EQ.26) GO TO 900
-  RUN27
      IF(IRUN.EQ.27) GO TO 900
-  RUN28
      IF(IRUN.EQ.28) GO TO 900
-  RUN29
      IF(IRUN.EQ.29) GO TO 900
-  RUN30
      IF(IRUN.EQ.30) GO TO 900
)0 CONTINUE
-  ALL RUNS
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 3 ENDS.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 4 STARTS:
---------------------------------------------------------
RITE GENERAL DATA ON TO THE GUSIE1.DTA TAPE, ETC...
      IF(SPDATA) CALL WRTSPC(LSPDA,NLSP,ISPDA,NISP,RSPDA,NRSP)
      IF(BLOCK) CALL WRTPOR(PE,PN,PH,PC,NX,NY,NZ,IPLANE)
  OLD PRACTICES RETAINED FOR REFERENCE:
      IF(SPDATA) CALL SPCDAT(IRUN)
      IF(BLOCK) CALL PORDAT(IRUN)
      IF(GRAPHS) CALL SORT(IRUN)
      IF(RESTRT) GO TO 902
    DO 901 INDVAR=1,25
      IF(IFIX(FIINIT(INDVAR)+0.1).NE.10101) GO TO 901
  CALL FLDDAT(IRUN)
  GO TO 902
1 CONTINUE
2 CALL DATAIO(WRT,10)
      IF(MONITR) CALL DATAIO(WRT,-6)
  CONTINUE
  STOP
  END
RECTIVE**FLDDAT
 *FILE NAME: MODFLD.FTN
 *ABSTRACT: SATELLITE MODEL FLDDAT SUBROUTINE.
 *DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983).
  SUBROUTINE FLDDAT(IRUN)
----------------------------------------------------------
  FIELDS IS USED TO SPECIFY NON-UNIFORM INITIAL FIELDS.
  EARTH SETS UNIFORM INITIAL FIELDS TO FIINIT(MPHI), EXCEPT
  WHEN FINIIT(MPHI)=10101.0 WHICH IS THE SIGNAL THAT EARTH
  READS THE NON-UNIFORM FIELDS TAPE THE CONTENTS OF WHICH
  ARE SET HERE.
  IT IS ESSENTIAL TO PROVIDE SETTINGS FOR ALL THOSE MPHI'S
  FOR WHICH FIINIT(MPHI) HAS BEEN SET TO 10101.THE SETTING
  ORDER  M U S T  FOLLOW THE STANDARD ORDER OF BLOCK DATA.
  NOTE: EARTH PRINTS OUT THE INITIAL FIELDS IF KOUTPT=-1.

  PLEASE NOTE THAT THE PHI ARRAY  M U S T BE SET AS
  PHI(IY,IX)=...,NOT AS PHI(IX,IY)=...
  ALSO, YOU  M U S T  DIMENSION PHI TO THE MAXIMUM
  DIMENSIONS USED IN THE RUNS DEFINED IN SATLIT, VIZ.
  DIMENSION PHI(NY MAX.,NX MAX.)
---------------------------------------------------------
TER 1     PRELIMINARIES
---------------------------------------------------------
CLUDE 9,CMNGUSSI.FTN/G
  LOGICAL LOGIC1,LOGIC
  DIMENSION LOGIC(100)
  COMMON/LDATA/LOGIC1(309)
  EQUIVALENCE (LOGIC(1),LOGIC1(210))
  DIMENSION INTGR(100)
  COMMON/IDATA/INTGR1(194)
  EQUIVALENCE (INTGR(1),INTGR1(95))
  DIMENSION RE(100)
  COMMON/RDATA/RE1(421)
  EQUIVALENCE (RE(1),RE1(322))
  COMMON/BOUND/LOCREG(60),
 &TR1,CP1R1(7),VP1R1(7),CP2R1(5),VP2R1(5),CPNR1(5),VPNR1(5),
 &TR2,CP1R2(7),VP1R2(7),CP2R2(5),VP2R2(5),CPNR2(5),VPNR2(5),
```

```
     &TR3,CP1R3(7),VP1R3(7),CP2R3(5),VP2R3(5),CPNR3(5),VPNR3(5),
     &TR4,CP1R4(7),VP1R4(7),CP2R4(5),VP2R4(5),CPNR4(5),VPNR4(5),
     &TR5,CP1R5(7),VP1R5(7),CP2R5(5),VP2R5(5),CPNR5(5),VPNR5(5),
     &TR6,CP1R6(7),VP1R6(7),CP2R6(5),VP2R6(5),CPNR6(5),VPNR6(5),
     &TR7,CP1R7(7),VP1R7(7),CP2R7(5),VP2R7(5),CPNR7(5),VPNR7(5),
     &TR8,CP1R8(7),VP1R8(7),CP2R8(5),VP2R8(5),CPNR8(5),VPNR8(5),
     &TR9,CP1R9(7),VP1R9(7),CP2R9(5),VP2R9(5),CPNR9(5),VPNR9(5),
     &TR10,CP1R10(7),VP1R10(7),CP2R10(5),VP2R10(5),CPNR10(5),VPNR10(5)
      DIMENSION TCVREG(350)
      EQUIVALENCE (TCVREG(1),TR1)
C$INCLUDE 9,GUSSEQUI.FTN/G
      LOGICAL CARTES,POLAR,SPDATA,SKEW,TWODYZ,ONEDZ,STOVAR(25),
     &   SOLVAR(25),PRINT(25),RESID(25),CMPRSS,CONEMU,LSP1,
     &   CONRHO,EMDOT,ONEPHS,INCORE(10),ISAVED,SAVEI,SAVEM,
     &   RESTRT,XCYCLE,MONITR,REGION(10),STEADY,WHOLEP,SLABPP,
     &   RAIN,BLOCKZ,PWSTAG,RUN(30),PLOT,RESMAP,FLAG,BLOCK,
     &   TEST,CONC1(4),DISTIL
      LOGICAL SPRESS,PARAB,DONACC,OVERLY,SACC,
     &   GUSSIE,CATHY,CONNIE,CORA,ESTER,FLASH,FLORA,
     &   FOCS,GENMIX,HESTER,PICALO,PLANT,SPLASH,HELGA,
     &   TACT,TIBALT,TOPSI,PAMELA,TABLES,WSTAG,
     &   CONMOD,GROSTA,SUBPST,SUBWGR
      EQUIVALENCE (CARTES,LOGIC1(1)),(POLAR,LOGIC1(2)),
     &   (SPDATA,LOGIC1(3)),(SKEW,LOGIC1(4)),(TWODYZ,LOGIC1(5)),
     &   (ONEDZ,LOGIC1(6)),(STOVAR(1),LOGIC1(7)),
     &   (SOLVAR(1),LOGIC1(32)),(PRINT(1),LOGIC1(57)),
     &   (RESID(1),LOGIC1(82)),(CMPRSS,LOGIC1(107)),
     &   (CONEMU,LOGIC1(108)),(LSP1,LOGIC1(109)),
     &   (CONRHO,LOGIC1(110)),(EMDOT,LOGIC1(111)),
     &   (ONEPHS,LOGIC1(112)),(INCORE(1),LOGIC1(113))
      EQUIVALENCE(ISAVED,LOGIC1(123)),(SAVEI,LOGIC1(124)),
     &   (SAVEM,LOGIC1(125)),(RESTRT,LOGIC1(126)),
     &   (XCYCLE,LOGIC1(127)),(MONITR,LOGIC1(128)),
     &   (REGION(1),LOGIC1(129)),(STEADY,LOGIC1(139)),
     &   (WHOLEP,LOGIC1(140)),(SLABPP,LOGIC1(141)),
     &   (RAIN,LOGIC1(142)),(BLOCKZ,LOGIC1(143)),
     &   (PWSTAG,LOGIC1(144)),(RUN(1),LOGIC1(145)),
     &   (PLOT,LOGIC1(175)),(RESMAP,LOGIC1(176)),
     &   (FLAG,LOGIC1(177)),(BLOCK,LOGIC1(178)),(TEST,LOGIC1(179))
      EQUIVALENCE(CONC1(1),LOGIC1(206))
      EQUIVALENCE (SPRESS,LOGIC(100)),(PARAB,LOGIC(99)),
     &   (DONACC,LOGIC(98))
      EQUIVALENCE (OVERLY,LOGIC(96)),(SACC,LOGIC(95)),
     &   (WSTAG,LOGIC(94))
      EQUIVALENCE (DISTIL,LOGIC(90))
      EQUIVALENCE (PAMELA,LOGIC(67)),(TOPSI,LOGIC(66)),
     &   (TIBALT,LOGIC(65)),(TACT,LOGIC(64)),(HELGA,LOGIC(63)),
     &   (SPLASH,LOGIC(62)),(PLANT,LOGIC(61)),(PICALO,LOGIC(60)),
     &   (HESTER,LOGIC(59)),(GENMIX,LOGIC(58)),(FOCS,LOGIC(57)),
     &   (FLORA,LOGIC(56)),(FLASH,LOGIC(55)),(ESTER,LOGIC(54)),
     &   (CORA,LOGIC(53)),(CONNIE,LOGIC(52)),(CATHY,LOGIC(51)),
     &   (GUSSIE,LOGIC(50))
      EQUIVALENCE (TABLES,LOGIC(48)),(CONMOD,LOGIC(47)),
     &   (GROSTA,LOGIC(46)),(SUBPST,LOGIC(45))
      EQUIVALENCE (SUBWGR,LOGIC(43))
      INTEGER FSTEP,FSWEEP,TSTSWP,ITAB(8),MTABVR(8)
      DIMENSION ISPCSO(25),LITER(25)
      EQUIVALENCE (NX,INTGR1(1)),(NY,INTGR1(2)),
     &   (NZ,INTGR1(3)),(ISPCSO(1),INTGR1(4)),
     &   (NREGN,INTGR1(29)),(NPHI,INTGR1(30)),
     &   (LITKE,INTGR1(31)),(LITHYD,INTGR1(32)),
     &   (LITH,INTGR1(33)),(LITCNC,INTGR1(34)),
     &   (LITSLB,INTGR1(35)),(NRUN,INTGR1(36)),
     &   (LITER(1),INTGR1(37)),(FSTEP,INTGR1(62)),
     &   (FSWEEP,INTGR1(63)),(LSTEP,INTGR1(64))
      EQUIVALENCE (LSWEEP,INTGR1(65)),(NPRINT,INTGR1(66)),
     &   (IERRP,INTGR1(67)),(IMAXP,INTGR1(68)),
     &   (IEMU1,INTGR1(69)),(IXMON,INTGR1(70)),
     &   (IYMON,INTGR1(71)),(IZMON,INTGR1(72)),
     &   (KEMU,INTGR1(73)),(KMAIN,INTGR1(74)),
     &   (KINDEX,INTGR1(75)),(KGEOM,INTGR1(76)),
     &   (KINPUT,INTGR1(77)),(KSODAT,INTGR1(78))
```

```
 EQUIVALENCE(KFLAG,INTGR1(79)),(KCOMPF,INTGR1(80)),                     MEL07210
&   (KSORCE,INTGR1(81)),(KLES1D,INTGR1(82)),                            MEL07220
&   (KLES2D,INTGR1(83)),(KLES3D,INTGR1(84)),                            MEL07230
&   (KCOMPP,INTGR1(85)),(KADJST,INTGR1(86)),                            MEL07240
&   (KFLUX,INTGR1(87)),(KSHIFT,INTGR1(88)),                             MEL07250
&   (KOUTPT,INTGR1(89)),(KDIF,INTGR1(90)),                              MEL07260
&   (KCOMPU,INTGR1(91)),(KCOMPV,INTGR1(92)),                            MEL07270
&   (KCOMPW,INTGR1(93)),(KCOMPR,INTGR1(94))                             MEL07280
 EQUIVALENCE (KELIN,INTGR(100)),(MEANDF,INTGR(99)),                     MEL07290
&   (NUMCLS,INTGR(98)),                                                 MEL07300
&   (IRHO1,INTGR(97)),(IRHO2,INTGR(96)),                                MEL07310
&   (IZW1,INTGR(95)),(IZW2,INTGR(94)),                                  MEL07320
&   (MGRID,INTGR(93)),(KWALL,INTGR(92)),                                MEL07330
&   (IZPR1,INTGR(91)),(IZPR2,INTGR(90)),                                MEL07340
&   (ISTPR1,INTGR(89)),(ISTPR2,INTGR(88)),                              MEL07350
&   (NTPRIN,INTGR(87))                                                  MEL07360
 EQUIVALENCE (IMDOT,INTGR(86)),(IHSAT,INTGR(85)),                       MEL07370
&   (ICFIP,INTGR(84)),(NTABLE,INTGR(83)),(NTABVR,INTGR(82)),            MEL07380
&   (LINTAB,INTGR(81)),(NPRTAB,INTGR(80)),(NMON,INTGR(79)),             MEL07390
&   (ITAB(1),INTGR(71)),(MTABVR(1),INTGR(63)),                          MEL07400
&   (NZPRIN,INTGR(62)),                                                 MEL07410
&   (NXPRIN,INTGR(61)),(NYPRIN,INTGR(60)),                              MEL07420
&   (LDISTL,INTGR(59)),(TSTSWP,INTGR(58)),                              MEL07430
&   (KDBEXP,INTGR(57)),(KDBRHO,INTGR(56)),                              MEL07440
&   (KDBMDT,INTGR(55))                                                  MEL07450
 EQUIVALENCE (ILOOP1,INTGR(53)),(ILOOPN,INTGR(52)),                     MEL07460
&   (IPBP,INTGR(51)),(IZPRF,INTGR(50)),                                 MEL07470
&   (IZPRL,INTGR(49)),(ISTPRF,INTGR(48)),                               MEL07480
&   (ISTPRL,INTGR(47)),(KDBGEN,INTGR(46)),                              MEL07490
&   (IVELF,INTGR(45)),(IVELL,INTGR(44)),                                MEL07500
&   (IKEF,INTGR(43)),(IKEL,INTGR(42)),                                  MEL07510
&   (IENTF,INTGR(41)),(IENTL,INTGR(40)),                                MEL07520
&   (ICNCF,INTGR(39)),(ICNCL,INTGR(38)),                                MEL07530
&   (NVEL,INTGR(37)),(NKE,INTGR(36)),                                   MEL07540
&   (NENT,INTGR(35)),(NCNC,INTGR(34)),                                  MEL07550
&   (NZSTP,INTGR(33)),(NPRMNT,INTGR(32))                                MEL07560
 DIMENSION SPARE1(20),XUDIST(30),YVDIST(30),ZWDIST(50),                 MEL07570
&   SIGMA(25),CRIT(25),DTFALS(25),RESREF(25),                           MEL07580
&   TITLE(25),FIINIT(25),TFRAC(30)                                      MEL07590
 EQUIVALENCE (SPARE1(1),RE1(1)),                                        MEL07600
&   (TFRAC(1),RE1(21)),(XULAST,RE1(51)),                                MEL07610
&   (YVLAST,RE1(52)),(ZWLAST,RE1(53)),                                  MEL07620
&   (XUDIST(1),RE1(54)),(YVDIST(1),RE1(84)),                            MEL07630
&   (ZWDIST(1),RE1(114)),(SIGMA(1),RE1(164)),                           MEL07640
&   (CRIT(1),RE1(189)),(DTFALS(1),RE1(214)),                            MEL07650
&   (RESREF(1),RE1(239)),(EMU1,RE1(264)),                               MEL07660
&   (RHO1,RE1(265)),(RHO2,RE1(266)),                                    MEL07670
&   (TLAST,RE1(267)),(CFIPS,RE1(268))                                   MEL07680
 EQUIVALENCE (AMDOT,RE1(269)),(FIINIT(1),RE1(270)),                     MEL07690
&   (RELAXP,RE1(295)),(TITLE(1),RE1(296)),                              MEL07700
&   (DT,RE1(321)),                                                      MEL07710
&   (RINNER,ZWDIST(50)),(SNALFA,ZWDIST(49)),                            MEL07720
&   (PBAR,ZWDIST(48))                                                   MEL07730
 EQUIVALENCE (SLOEMU,RE(100)),(SLORHO,RE(99)),(RLXRHO,RE(98)),          MEL07740
&   (RHOMAX,RE(97)),(RHOMIN,RE(96)),                                    MEL07750
&   (EMUMAX,RE(95)),(EMUMIN,RE(94)),                                    MEL07760
&   (TKEMAX,RE(93)),(TKEMIN,RE(92)),                                    MEL07770
&   (RLXPZ,RE(91)),                                                     MEL07780
&   (ABUOY,RE(90)),(HREF,RE(89)),                                       MEL07790
&   (AGRAVX,RE(88)),(AGRAVY,RE(87)),                                    MEL07800
&   (AGRAVZ,RE(86)),(ATIME,RE(85))                                      MEL07810
 EQUIVALENCE (RLXPXY,RE(84)),                                           MEL07820
&   (EPSMAX,RE(83)),(EPSMIN,RE(82)),                                    MEL07830
&   (ARHO1,RE(81)),(BRHO1,RE(80)),(CRHO1,RE(79))                        MEL07840
 EQUIVALENCE (AZW2,RE(73)),(BZW2,RE(74)),                               MEL07850
&   (PINT,RE(75)),(CZW2,RE(76)),                                        MEL07860
&   (DZ,RE(78)),(ZWL,RE(77)),(ZW2M1T,RE(72)),                           MEL07870
&   (VELMIN,RE(70)),(VELMAX,RE(69)),                                    MEL07880
&   (FALCOM,RE(68)),(CMDOT,RE(67)),                                     MEL07890
&   (RLXMDT,RE(66)),(AMDTMX,RE(65)),(AMDTMN,RE(64)),                    MEL07900
&   (RADMAX,RE(63)),(RADMIN,RE(62)),                                    MEL07910
&   (PRESS,RE(61)),(CP1,RE(60)),                                        MEL07920
```

```
     &   (CP2,RE(59)),(CP3,RE(58)),(FSTOIC,RE(57)),
     &   (ARRCON,RE(56)),(PREEXP,RE(55)),(CA1I,RE(54))
      EQUIVALENCE (CA2I,RE(53)),(CA13,RE(52)),(CA23,RE(51)),
     &   (H1SAT,RE(50)),(H2SAT,RE(49)),
     &   (EMULAM,RE(48))
      EQUIVALENCE (TWPRCN,RE(42)),(PSATEX,RE(41))
      EQUIVALENCE(ELEXP,RE(39)),(EWALL,RE(38)),
     &   (SIGE,RE(37)),(SIGK,RE(36)),(TAUDK,RE(35))
      DIMENSION LOC1(6),LOC2(6),LOC3(6),LOC4(6),LOC5(6),
     &   LOC6(6),LOC7(6),LOC8(6),LOC9(6),LOC10(6)
      EQUIVALENCE (LOCREG(1),LOC1(1)),(LOCREG(7),LOC2(1))
      EQUIVALENCE (LOCREG(13),LOC3(1)),(LOCREG(19),LOC4(1))
      EQUIVALENCE (LOCREG(25),LOC5(1)),(LOCREG(31),LOC6(1))
      EQUIVALENCE (LOCREG(37),LOC7(1)),(LOCREG(43),LOC8(1))
      EQUIVALENCE (LOCREG(49),LOC9(1)),(LOCREG(55),LOC10(1))
      DIMENSION XUFRAC(30),YVFRAC(30),ZWFRAC(30),ENDIT(25)
      DIMENSION XFRAC(30),YFRAC(30),ZFRAC(30)
      EQUIVALENCE(XUDIST(1),XUFRAC(1),XFRAC(1)),
     &           (YVDIST(1),YVFRAC(1),YFRAC(1)),
     &            (ZWDIST(1),ZWFRAC(1),ZFRAC(1)),(LITCNC,LITC),
     &            (ENDIT(1),CRIT(1)),(RLXP,RELAXP)
      EQUIVALENCE (NPRMON,LITER(1))
      DIMENSION PHI(100,1),FDTAPE(3)
      INTEGER FTAPE
      LOGICAL FIRST
      INTEGER P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,EP,H1,H2,H3,C1,C2,
     &C3,C4
      DATA P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,KE,EP,H1,H2,H3,C1,C2,
     &C3,C4/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20/
      DATA FIRST,FTAPE,FDTAPE/.TRUE.,9,4HFIEL,4HDS.D,2HTA/
         IF(FIRST) CALL TAPES(FTAPE,FDTAPE,3,1,4*NX*NY)
      FIRST=.FALSE.
C---------------------------------------------------------------
CHAPTER 2  USER CHAPTER FOR SETTING INITIAL FIELDS.
C---------------------------------------------------------------
      NNZ=NZ
         IF(PARAB) NNZ=1
       DO 2000 IZ=1,NNZ
       DO 2000 MPHI=1,25
         IF(MPHI.EQ.2) GO TO 2000
         IF(IFIX(FIINIT(MPHI)+0.1).NE.10101) GO TO 2000
         IF(.NOT.STOVAR(MPHI).AND..NOT.SOLVAR(MPHI)) GO TO 2000
C************USER SECTION STARTS HERE************
C     FOR EACH VARIABLE, MVAR, FOR WHICH FIINIT(MVAR) IS SET TO
C     10101.0 IN BLOCK DATA, THE USER MUST INSERT HERE FORTRAN
C     STATEMENTS FOR SETTING HIS WANTED NON-UNIFORM INITIAL FIELD.
C     THE FORTRAN-STATEMENT MODEL TO BE FOLLOWED BY THE USER FOR
C     EACH MVAR IS AS FOLLOWS...
C
C        IF(MPHI.NE.MVAR) GO TO 205
C      DO 201 IX=1,NX
C      DO 201 IY=1,NY
C 201 PHI(IY,IX)=.........USER INSERTS FIELD VALUE AT IX,IY,IZ HERE.
C     WRITE(FTAPE) ((PHI(IY,IX),IY=1,NY),IX=1,NX)
C 205 CONTINUE
C
C     INSERT HERE ADDITIONAL FORTRAN SEQUENCES, SIMILAR TO THAT
C     GIVEN ABOVE, FOR EACH MVAR FOR WHICH FIINIT(MVAR)=10101.0
C************USER SECTION ENDS HERE  ************
                WRITE(6,*) '****** BEGINNING ',MPHI
      IF (MPHI.NE. 5) GOTO 210
       DO 201 IY=1,25
  201 PHI(IY,1)=0.0
      DO 202 IY=26,28
  202 PHI(IY,1)=0.0375
      DO 203 IY=29,100
  203 PHI(IY,1)=0.00001
      WRITE(FTAPE)(PHI(IY,1),IY=1,100)
                WRITE(6,*) '------ WRITTEN   ',MPHI
      GOTO 270
  210 IF (MPHI.NE. 9) GOTO 230
      DO 211 IY=1,100
```

```
211 PHI(IY,1)=1.0                                                    MEL08650
    WRITE(FTAPE)(PHI(IY,1),IY=1,100)                                 MEL08660
              WRITE(6,*) '------ WRITTEN   ',MPHI                    MEL08670
    GOTO 270                                                         MEL08680
230 IF ( MPHI .NE. 14 ) GOTO 240                                     MEL08690
    DO 213 IY=1,25                                                   MEL08700
213 PHI(IY,1)=1.58E06                                                MEL08710
    PHI(26,1)=3.486E06                                               MEL08720
    PHI(27,1)=6.670E06                                               MEL08730
    PHI(28,1)=6.770E06                                               MEL08740
    PHI(29,1)=6.3E06                                                 MEL08750
    PHI(30,1)=4.225E06                                               MEL08760
    PHI(31,1)=2.22E06                                                MEL08770
    PHI(32,1)=2.37E05                                                MEL08780
    PHI(33,1)=-1.72E06                                               MEL08790
    PHI(34,1)=-3.2E06                                                MEL08800
    PHI(35,1)=-6.57E06                                               MEL08810
    PHI(36,1)=-7.03E06                                               MEL08820
    DO 216 IY=37,100                                                 MEL08830
216 PHI(IY,1)=-1.28E07                                               MEL08840
    WRITE(FTAPE)(PHI(IY,1),IY=1,100)                                 MEL08850
              WRITE(6,*) '------ WRITTEN   ',MPHI                    MEL08860
    GOTO 270                                                         MEL08870
240 IF ( MPHI .NE. 17 ) GOTO 250                                     MEL08880
    DO 217 IY=1,25                                                   MEL08890
217 PHI(IY,1)=1.0                                                    MEL08900
    PHI(26,1)=0.9                                                    MEL08910
    PHI(27,1)=0.8                                                    MEL08920
    PHI(28,1)=0.7                                                    MEL08930
    PHI(29,1)=0.6                                                    MEL08940
    PHI(30,1)=0.5                                                    MEL08950
    PHI(31,1)=0.4                                                    MEL08960
    PHI(32,1)=0.3                                                    MEL08970
    PHI(33,1)=0.2                                                    MEL08980
    PHI(34,1)=0.1                                                    MEL08990
    PHI(35,1)=1.E-10                                                 MEL09000
    DO 219 IY=36,100                                                 MEL09010
219 PHI(IY,1)=1.E-10                                                 MEL09020
    WRITE(FTAPE)(PHI(IY,1),IY=1,100)                                 MEL09030
              WRITE(6,*) '------ WRITTEN   ',MPHI                    MEL09040
    GOTO 270                                                         MEL09050
250 IF ( MPHI .NE. 18 ) GOTO 260                                     MEL09060
    DO 251 IY=1,25                                                   MEL09070
251 PHI(IY,1)=1.E-10                                                 MEL09080
    PHI(26,1)=0.05                                                   MEL09090
    PHI(27,1)=0.155                                                  MEL09100
    PHI(28,1)=0.26                                                   MEL09110
    PHI(29,1)=0.365                                                  MEL09120
    PHI(30,1)=0.47                                                   MEL09130
    PHI(31,1)=0.575                                                  MEL09140
    PHI(32,1)=0.68                                                   MEL09150
    PHI(33,1)=0.785                                                  MEL09160
    PHI(34,1)=0.89                                                   MEL09170
    PHI(35,1)=1.0                                                    MEL09180
    DO 253 IY=36,100                                                 MEL09190
253 PHI(IY,1)=1.0                                                    MEL09200
    WRITE(FTAPE)(PHI(IY,1),IY=1,100)                                 MEL09210
              WRITE(6,*) '------ WRITTEN   ',MPHI                    MEL09220
    GOTO 270                                                         MEL09230
260 IF(MPHI .NE. 19 ) GOTO 270                                       MEL09240
    DO 261 IY=1,25                                                   MEL09250
261 PHI(IY,1)=1.E-10                                                 MEL09260
    PHI(26,1)=0.05                                                   MEL09270
    PHI(27,1)=0.045                                                  MEL09280
    PHI(28,1)=0.04                                                   MEL09290
    PHI(29,1)=0.035                                                  MEL09300
    PHI(30,1)=0.03                                                   MEL09310
    PHI(31,1)=0.025                                                  MEL09320
    PHI(32,1)=0.02                                                   MEL09330
    PHI(33,1)=0.015                                                  MEL09340
    PHI(34,1)=0.01                                                   MEL09350
    PHI(35,1)=1.E-10                                                 MEL09360
```

```
      DO 262 IY=36,100
  262 PHI(IY,1)=1.E-10
      WRITE(FTAPE)(PHI(IY,1),IY=1,100)
                WRITE(6,*) '------ WRITTEN    ',MPHI
  270 CONTINUE
 2000 CONTINUE
      RETURN
      END
```

REFENCES

1. Shavit Z., "A Transient One-Dimensional Model for Metal Combustion in Steam," Contractor Report NPS72-86-001CR , Naval Postgraduate School, September 1986.
2. Spalding D. B., "A General Purpose Computer Program for Multi-dimensional One and Two-Phase Flow," Mathematics and Compuers in Simulation, North Holland Press, Vol XXIII, pp 257-276, 1981.
3. Patankar S. V., "Numerical Heat Transfer and Fluid Flow," Hemisphere Publishing Corporation, 1980.

DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2
3.	Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93943-5100	1
4.	Distinguished Professor Allen E. Fuhs Code 67Fu Naval Postgraduate School Monterey, California 93943-5100	2
5.	Mr. Donald E. Phillips, Code R10A White Oak Laboratory Naval Surface Weapons Center Silver Springs, Maryland 20910	3
6.	Dr. Franklin D. Hains, Code R14 White Oak Laboratory Naval Surface Weapons Center Silver Springs, Maryland 20910	1
7.	Mr. Wayne K. Reed, Code R14 White Oak Laboratory Naval Surface Weapons Center Silver Springs, Maryland 20910	1
8.	Information Services Division, Code X20 White Oak Laboratory Naval Surface Weapons Center Dahlgren, Virginia 22448	1
9.	Mr. George Daniello White Oak Laboratory Naval Surface Weapons Center Silver Springs, Maryland 20910	1
10.	Professor Oscar Biblarz Code 67Bi Department of Aeronautics Naval Postgraduate School Monterey, CA 93943-5100	1

11. Mr. Francis J. Romano, SEA 63R3 1
Naval Sea Systems Command
Washington, DC 20360

12. Dr. William Sykes, Code 175 1
David W. Taylor Naval Ship Research
and Development Center
Bethesda, Maryland 20084

13. Mr. Raymond P. Gogolewski 1
Defense Advanced Research Project Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

14. Mr. Charles Beatty 1
Naval Undersea Center
San Diego, California 92132

15. Jonathan L. Minner 1
Naval Surface Weapons Center
Dahlgren, Virginia 22448

16. Zernow Technical Services 1
Attn: Louis Zernow
425 West Bonita
San Dimas, California 91773

17. Dr. Louis Baker, Jr. 1
Argonne National Laboratories
9700 South Cass Avenue
Building 207
Argonne, Illinois 60439

18. R. G. S. Sewell 1
Code 3835
Naval Weapons Center
China Lake, California 93555

19. Professor David Salinas 1
Code 69Sa
Departwent of Mechanical Engineering
Naval Postgraduate Schhol
Monterey, CA 93943-5100

20. R. R. Durrell 1
Code R12
Naval Surface Weapons Center
Silver Springs, Maryland 20910

21. D. R. Kennedy Associates, Inc. 1
4940 El Camino Real, Suite 209
Post Office Box 4003
Mountain View, California 94040

22. Mr. J. M. McNerney 1
Battelle
Columbus Laboratories
505 King Avenue
Columbus, Ohio 43201

23. Dr. G. E. Jensen 1
Chemical Systems Division
United Technology Corporation
Post Office Box 358
Sunnyvale, California 94086

24. Dr. Edward G. Liszka 1
Mark 50 Torpedo Office
Naval Sea Systems Command
Washington, DC 20360

25. Mr. Zeev Shavit (48) 2
Ministry of Defense
P.O. Box 2250
Haifa, Israel

26. Professor A. Gany 1
Department of Aeronautical Engineering
Technion - Israel Institute of Technology
Haifa, Israel

27. Research Administration 1
Code 012A
Naval Postgraduate School
Monterey, CA 93943-5100